Ref. Ares(2021)2891271 - 30/04/2021

DIGITAL MANUFACTURING PLATFORMS FOR CONNECTED SMART FACTORIES

D2.10 QU4LITY Digital Models and Vocabularies

Deliverable Id:	**D2.10**
Deliverable Name:	**QU4LITY Digital Models and Vocabularies**
Status:	**Final**
Dissemination Level:	**PU**
Due date of deliverable:	**31/03/2021**
Actual submission date:	**30/04/2021**
Work Package:	**WP2**
Organization name of lead contractor for this deliverable:	**EPFL**
Author(s):	**Xiaochen Zheng, Kiritsis Dimitris, Joanna Wortel, Jeroen Greidanus, Ana Teresa Correia, Sebastian Scholze**
Partner(s) contributing:	**SIEMENS, PHILIPS, PRIMA, TID, TTS, FHG, EPFL, ATB**

Abstract: This deliverable reviews a range of data standards, models and vocabularies that drive the flow and exchange of digital data across different ZDM equipment and processes. The specification of data models will be based on existing standards for representing plants, production processes and quality processes information. Principles and methodology for using such specifications are also explained in this deliverable. The models are applied to multiple pilots as proof-of-concept.

Co-funded by the Horizon 2020 Programme
of the European Union

Contents

HISTORY

Version	Date	Modification reason	Modified by
0.1	25/03/2021	First version	Xiaochen Zheng (EPFL)
0.2	19/04/2021	Internal Peer review	Daniel García (TID)
0.3	20/04/2021	Revised version addressed reviewer's comments	Xiaochen Zheng (EPFL)
1.0	30/04/2021	Final version for submitting	Xiaochen Zheng (EPFL)

1. Executive summary

This deliverable refers to the QU4LITY project Work Package (WP) 2 Task T2.5. The topic is "Specification and Prototyping of Digital Models, Vocabularies and Digital Twins". This task is responsible for:

- Requirements analysis for the digital representation
  - Pilot plant data (eventually extended to supply chain data),
  - Information flow exchange through Zero Defect Manufacturing (ZDM) components and digital platforms.
- Specification of models and vocabularies for Digital Twins applications to ease
  - Data interoperability,
  - Simulation and implementation.
- Ensure
  - Compliance to existing standards.
  - Openly accessible (open source) digital models and tools.

The relations of this task with other tasks in WP2 are shown in Figure 1.

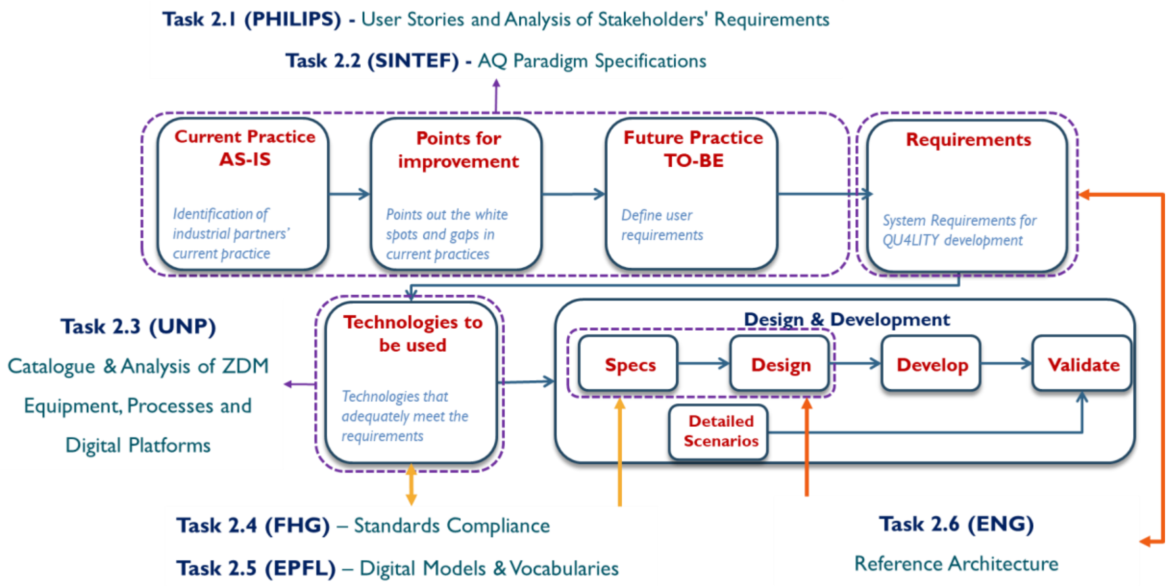

Figure 1 Overview of WP2 tasks and their relations

In this report, we briefly introduce the background of this task under the QU4LITY AQ vision as well as in the reference architecture. Following the principles and methodology that were defined in our previous report (Deliverable D2.9), we analyze the user stories and stakeholders' requirements, relevant existing top-level and domain ontologies thus to define the QU4LITY vocabularies. Moreover, based on the MPFQ model and Z-BREAK model from previous projects, we design the data models for AQ scenario including the RMPFQ model and the semantic-driven digital twin model. To demonstrate the semantic-driven approach, we developed multiple ontologies covering top-level, domain-level and application-level ontologies corresponding to several process-oriented and machine-oriented pilots.

The rest of the deliverable is structured as follows. Section 2 briefly introduces the background of QU4LITY project task T2.5 and some key concepts related to this task, such as Autonomous Quality (AQ), semantic modelling, data interoperability etc. Section 3 comprehensively reviews the state-of-the-art of existing industrial data standards; the methodology of ontology engineering and semantic modelling, and several important standards and languages used for semantic modelling. Some relevant domain ontologies are also reviewed in this section. Section 4 explains the principles and methodology to be used in this project for the development of ontology-based models for different pilots, including competency questions, specification and temples of vocabularies, potential data models to be referred to in future etc. Section 5 presents the competency questions and specification of vocabularies for QU4LITY projects. Section 6 introduces the data models and their applications based on multiple pilots. Section 7 describes the semantic modelling results including the top-level ontology, domain ontology and multiple application-level ontologies. Section 8 concludes this deliverable and introduces the future works which will be involved in other tasks.

2. Introduction

2.1 Background and objectives

One of the core concepts of QU4LITY project is ZDM. It is a paradigm that aspires to develop methodologies, technologies and integrated tools for maintenance, quality control, and logistics of production that takes advantage of the knowledge of the process and the system. It aims to reduce defects as much as possible thanks to the implementation of preventive actions. Industry 4.0 has reshaped the modern manufacturing systems. New technologies, platforms and data spaces are networked to create value by generating, analyzing and communicating data seamlessly. In light of this, for companies to achieve ZDM, operations and products must be smart and connected. The digitalization of manufacturing systems allows access to data by implementation of Cyber-physical Production Systems (CPSs) and generates connectivity and interoperability through the Industrial Internet of Things (IIoT). To this end, the interaction between hardware/software and data management makes the ZDM concept easier to be implemented due to the availability of the required amount of data for advanced technologies such as machine learning to work properly [1].

An AQ paradigm for ZDM in a connected smart Factory 4.0 requires implementation of many interrelated control loops (self-adaptive real-time in-process, deep planning, simulation-based fault prognosis, human in the loop) for real-time adaptation, flexible composition, smart planning and continuous learning. The cost-effective implementation of such control loops demands not only massive digitization of the shop floor, advanced data analytic and storage, but also requires interoperable dedicated peer-to-peer industrial data spaces, easier equipment and digital manufacturing platform services orchestration, etc.

The digital platforms of QU4LITY project for implementing the AQ concept will adhere to well-defined specifications regarding the digital representation of plants and supply chain processes, as well as of the digital data exchanged in the scope of the ZDM equipment and digital platforms operation. These specifications will ease data interoperability, simulation and implementation of digital twins' applications.

Despite the clear definition of system functions and requirements, data interoperability remains one of the grand challenges when reengineering information systems. Data integration is a difficult task since data source can be heterogeneous in syntax, schema, or semantics. In order to achieve semantic interoperability in a heterogeneous information system, the meaning of the integrated data must be understood and shared among the actors of the system. As semantics has to do with the study of meanings, semantic interoperability occurs when different systems, such as software tools, interacting with each other or with people, can interact and make effective use of the terms that are used in the interaction.

Semantic interoperability is understood as the ability of information systems to exchange data unambiguously with shared meaning; therefore, it is a requirement that enables machine computable logic, inference, knowledge discovery, and data federation between information systems. Semantic interoperability is concerned with the packaging of data (syntax) as well as with the simultaneous transmission of the meaning with the data (semantics). This is accomplished by adding data about the data, called metadata, linking each data element to a controlled, shared vocabulary. The meaning of the data is transmitted with the data itself, in an information package that is independent of any information system. It is this shared vocabulary, and its associated links to an ontology, which provides the foundation and capability of machine interpretation, inferences, and logic.

A range of data models and vocabularies are required to drive the flow and exchange of digital data across different ZDM equipment and processes. The specification of these data models should be based on existing standards for representing plants, production processes and quality processes information, such as B2MML, AutomationML, CAEX, PLCOpen, COLLADA, MTConnect, MIMOSA and more. The specifications of various openly accessible digital models, such as MPFQ-model (Material, Production Process, Product Functions/Features, Product Quality), will provide reference mechanisms (e.g., APIs and tools for Create-Read-Update-Delete (CRUD) operations with bindings in different languages and formats) for implementing similar models for QU4LITY project.

The main objectives of this deliverable are:

- Definition of principles and methodology for developing ontology-based digital models which will be implemented in different QU4LITY project pilots,
- Definition of scope and granularity of the ontological models and collection of competency questions and vocabularies based on previous/ongoing relevant projects like Industrial Ontologies Foundry (IOF) ontologies, Z-Fact0r ontology, Boost 4.0 vocabularies etc,
- Definition of the top-terms and vocabularies based on autonomous quality vision and specifications, as well as user stories and stakeholders' requirements,
- Specification of the data models and their applications in machine-oriented and process-oriented pilots,
- Semantic models and their applications including the top-level ontology, domain-level ontology and multiple application-level ontologies,
- Digital twin modelling based on the QU4LITY data models and integrate them with semantic modelling to enable the novel Cognitive twin concept.

2.2 QU4LITY vision and specifications for digital modelling

Correctly understand the QU4LITY vision and specifications is one of the critical preconditions for defining digital models of QU4LITY. The aim of QU4LITY project is to create an open, certifiable and highly standardized, SME-friendly and transformative shared data-driven ZDM product and service model. It will demonstrate how European industry can build unique and highly tailored ZDM strategies and competitive advantages through an orchestrated open platforms ecosystem, ZDM atomized components and digital enablers.

QU4LITY Task 2.2 has thoroughly investigated the QU4LITY VISION and AQ paradigm specifications. As specified in its final report (deliverable D2.4, "QU4LITY Vision & Autonomous Quality (AQ) Model"), AQ is intended as a paradigm for ZDM in a connected smart Factory 4.0, which requires the implementation of interrelated control loops for real-time adaptation, flexible composition, smart planning and continuous learning. AQ can be defined as a real-time quality control process supported by Industry 4.0 enabling technologies where, at the maximum level of system autonomy, the decisions (closing loop) are taken by software after a deep data analysis. AQ aims at reducing the human input in the data analysis and process control to achieve the automation of the loops of information through improved use of more complex control systems. The goal is to achieve autonomous decision-making processes to assure the quality of production processes and related output in an autonomous way.

In an industry 4.0 perspective, AQ in all production steps can be a challenging task. To reach zero-defect in different process steps by optimizing both equipment or production processes, a fuzzy area of how to tackle predictive and prescriptive interaction of cyber physical production systems (CPPS) and full automation for production lines needs to be implemented and controlled. Those autonomous manufacturing processes will see and deploy different levels of human-AI collaboration. According to Industry 4.0 classification, QU4LITY will address the implementation of processes and manufacturing equipment up to Autonomy Level 4, where processes will run autonomously, and the human is only involved to supervise and intervene in emergency situations. The QU4LITY AQ model will meet the Industry 4.0 ZDM challenges (cost and time effective brownfield ZDM deployment, flexible ZDM strategy design & adaptation, agile operation of zero-defect processes & products, zero break down sustainable manufacturing process operation and human-centered manufacturing).

One of the main challenges of AQ is the management of the complex and heterogenous data and information of manufacturing systems. As described in the QUALITY Vision report (D2.4), AQ relies on Industry 4.0 technology and devices, autonomously communicating with each other along value chains. The integration of process and parts monitoring and control along the value chain can enhance traceability and earlier detection enabling quality improvements and defect reductions. Nowadays, in the industrial big data era, a lot of data is available, including sensor readings, inspection measurements, optical images as well as structured/unstructured data sources. This multi-source data is becoming an indispensable resource for production managements and quality improvement. Complexity of the quality control rises also since there is a need to go beyond one manufacturing process quality control. The measurement data contains a large amount of manufacturing process and product information. Technological means to control quality of the manufacturing process include also perception of the environment (i.e. tracking operator movements), feedback from the operators or machines, and external information collected from web, customers, product use phase and supply chains. At the same time, the modelling technology of multi-station manufacturing system is being improved continuously. Together with the different data sources available, the manufacturing process modelling techniques provide great potential for root cause identification of manufacturing process failures.

Therefore, Industry 4.0 offers opportunities in quality management: reduction in complexities, costs, risks, waste, dependency, vagueness and increase simplicity, convenience, interconnectivity, flexibility, productivity human-machine collaboration. Nevertheless, it also challenges to overcome technological complexity, the need for new skills as well as the adoption of new technologies. To better manage the complex industrial data, at least the following data management tasks must be addressed:

1) Data acquisition and sensing: Data acquisition (Data in Motion Models and Services for Industrial Internet of Things), data protection (Data anonymization, confidentiality, encryption and privacy preservation services),
2) Data processing and analysis: data storage (Data Spaces, Data Lake, Linked Data, Distributed Storage, Knowledge representation services), data analytics (Semantic analysis, Data discovery, Advanced Data Analytics (Edge Analytics, Cloud Analytics) services),
3) Decision support: Cognition, Prediction and prescription, Simulation, Machine Learning, Reinforcement.

To cope with this challenge, the Task 2.5 "Specification and Prototyping of Digital Models, Vocabularies and Digital Twin" aims to create digital models to empower the data management of QU4LITY paradigm. The digital model combined with semantic engineering technologies will help realize the above-mentioned targets.

2.3 Role of Digital Models and Vocabularies in QU4LITY Reference Architecture

The QU4LITY Reference Architecture (Q-RA), as shown in Figure 2, is the conceptual framework that is going to drive the design and the implementation of any QU4LITY-based solution. As every RA, its primary goal is to present, in a coherent and homogenous way, the underlying integration principles and digital technologies to be adopted in order to implement our AQ vision, where real-time quality control processes (supported by Industry 4.0 enabling technologies) provide maximum level of system autonomy based on closed-loop decisions. More details about QU4LITY Reference Architecture are introduced in deliverable D2.12 "Reference Architecture and Blueprints".

The digital models and vocabularies play an important role in the Q-RA to enhance interoperability between different components. The information is generated by using, monitoring, controlling and analyzing connected entities and sub-systems, remaining within a "domain" or being exchanged between "domains". Both raw and processed information is used by the different ZDM services and applications to fulfil intended task for a given activity in the system.

Digital Models and Vocabularies: Sharing digital models and vocabularies provides the capability to exchange information in the whole system with a common interpretation of information. In this contest, basically two levels of data interoperability are considered: syntactic interoperability is to exchange information in a common data format with a common protocol to structure the data; and semantics interoperability is to interpret the meaning of the symbols in the messages correctly. These interoperability components provide a flexible method of composing services so that the system behavior can be adapted at run-time to enable advanced ZDM processes.

As shown in Figure 2, the Digital Models and Vocabularies layer has close connections with the Interoperability Assurance Layer. It takes input from Interoperability Assurance Layer about user stories and stakeholder’s requirements for assuring interoperability among components. In return, it provides semantic models to enable the interoperability among ZDM components and digital platforms. Besides, it also provides data models and semantic models to support data-driven services and digital twin modelling components.

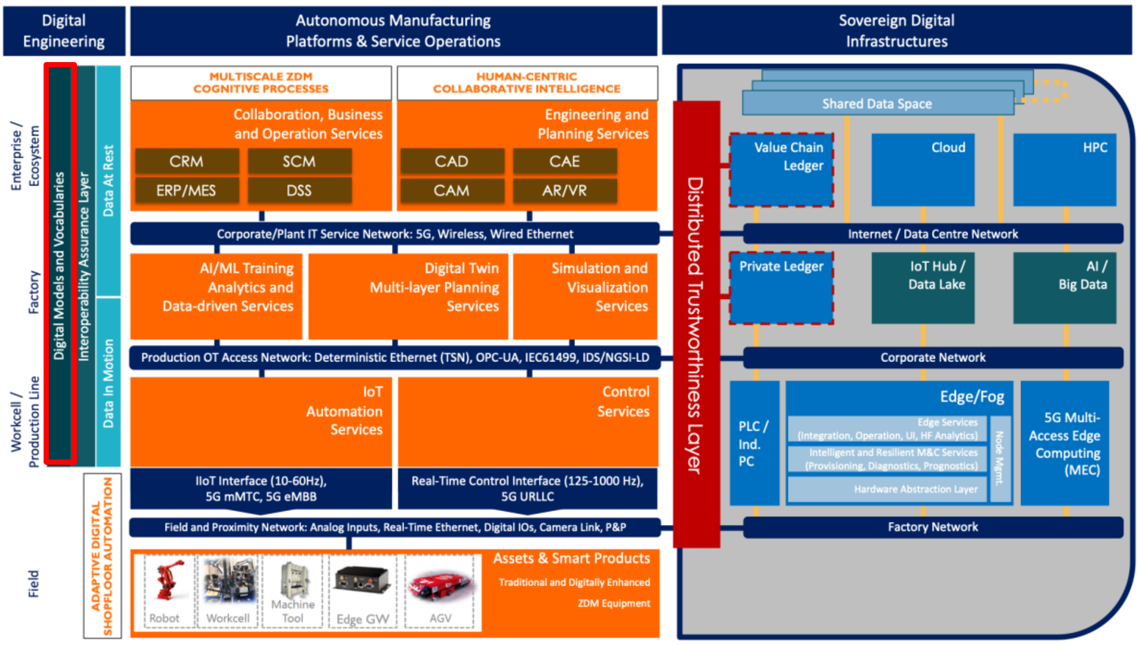

Figure 2 QU4LITY Reference Architecture

3. Principles and methodology

In our previous report (Deliverable D2.9), we have thoroughly reviewed the state-of-the-art about existing industrial data exchange standards and protocols, as well as relevant theories about ontology engineering and semantic modelling. In this report we have updated the review results. The updated version is attached in **Appendix A**. Based on the review results, we defined our principles and methodology to guide our tasks.

3.1 Principles

Based on the review of previous research and relevant projects, a series of principles are defined. These principles are extracted from the best practices supported by the IOF group.

- Granularity: Clear definition of context and scope of the representation
- Selection of the formats (and related serialization)
- Analysis and reuse of the existing domain ontologies
- Provision of textual (or natural language) definitions for each entity
- Setting up of unique Identifiers and Naming Conventions for each new entity
- Provision of a logic to foster reasoning and machine-based inference
- Compliance with QU4LITY architecture standards, e.g. Reference Architecture Model for Industry 4.0 (RAMI4.0). RAMI 4.0 is a framework that enables standards to be identified in order to determine whether there is any need for additions and amendments. This model is complemented by the Industry 4.0 components. RAMI 4.0 brings together the most important aspects of Industry 4.0 and is considered as one of the central orientation guides for Industry 4.0. as it provides a comprehensive view of the industrial landscape in a broad sense. The RAMI 4.0 standard builds strongly on the concepts of the Industry 4.0 Component and its Asset Administration Shell (AAS).
  The QU4LITY task T2.2 (deliverable D2.4) has already discussed the structure of RAMI 4.0 and the compliance requirements for QU4LITY projects. Task 2.4 (deliverable D2.8) will specifically concentrate on interoperability and standards-related issues that service as a background for the reference architecture compliance analysis and conception of interoperability specifications. This deliverable (task T2.5) focuses on the specifications and prototyping of digital models for representing plants, production processes and quality processes information. In industry, the Key Performance Indicator (KPIs) are widely used to quantify the performance of quality in smart manufacturing systems. A series of KPIs have been or will be defined by task T2.2. The specifications and prototyping of digital models in this task have to take into consideration of these KPIs such as their formulas and data properties etc. Moreover, certain compliance requirements in ZDM context regarding the interoperability and standards-related issues have been defined by T2.4, including conformity, complexity and readability, consistency, harmonization, informational capacity, industrial application, and integrity etc.

These requirements should also be followed when defining specifications of digital models.

- BFO compliance: Creation of each new entity starting from BFO (through existing core models). Entities in the future QU4LITY semantic framework should be arranged based on BFO. Two varieties are included in BFO, including continuant comprehending entities, such as three-dimensional enduring objects, and occurrent comprehending processes conceived as extended through (or as spanning) time. Using BFO framework makes it possible to merge the other manufacturing domain ontology structured by BFO. Originated from BFO, ontology design principles can be summarized as follows:
    - use single nouns (except data) and avoid acronyms
    - ensure univocity of terms and relational expressions
    - distinguish the general from particular
    - provide all non-root terms with definitions
    - use essential features in defining terms and avoid circularity
    - start with the most general terms in the domain
    - use simpler terms than the term you are defining (to ensure intelligibility)
    - do not create terms for universals through logical combination
    - structure ontology around *is_a* hierarchy and ensure *is_a* completeness single inheritance

The general structure of the BFO methodology is shown in Figure 3:

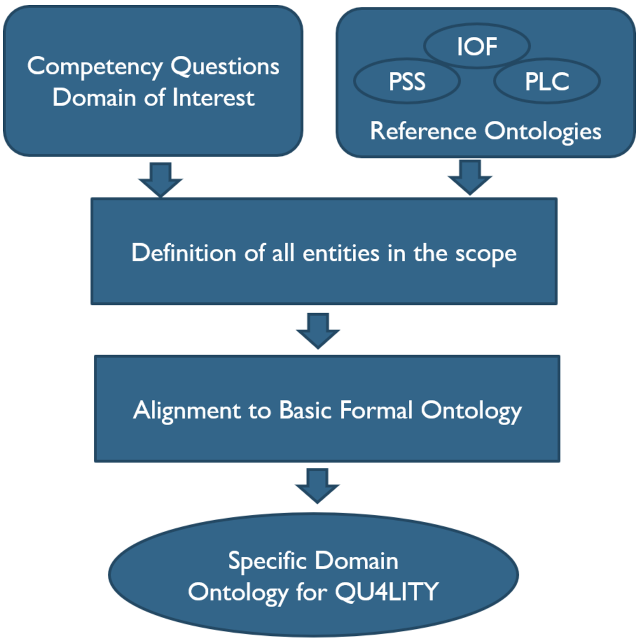

Figure 3 General structure of the BFO methodology.

- IOF-compliance: the design of the ontologies for QU4LITY follows the IOF architecture to facilitate future integration and interoperation with other domain ontologies. The details of applying IOF architecture is introduced in the Methodology section.

3.2 Methodology

- ### 3.2.1 IOF Ontology architecture

As mentioned in the principles, the development of QU4LITY ontologies follow the IOF architecture. The multi-layer IOF architecture, as shown in Figure 39, is created to facilitate a coherent and consistent development of different domain- and application-specific ontologies. It contains multiple layers of ontologies where the upper layers make use of the lower layers. It is expected that organizations will extend IOF ontologies into sub-domains and applications relevant to the organization. Starting from the most common or general (i.e., foundational, aka upper, ontology), to domain independent (reference) ontologies (e.g., reference ontologies for qualities, time, units of measure, etc.), to IOF domain reference ontologies, subdomain (i.e., more specific) ontologies, and finally application ontologies. The expectation is that this arrangement will allow IOF ontologies to be constructed cumulatively, built off lower layers. IOF ontologies are intended to be reference ontologies representing the more common or general notions that occur in a domain. In contrast to application ontologies that provide additional levels of detail, distinction, and specialization needed to represent a subdomain or meet more stringent requirements of an application or a specific usage environment.

Following the IOF architecture, we will use the IOF-core ontology as the top-level reference ontology. Then a domain specific reference ontology for QU4LITY will be developed aim at all QU4LITY pilots. Under that, two subdomain ontologies, i.e. process-oriented subdomain ontology and machine-oriented subdomain ontology will be developed corresponding to the two types of the QU4LITY pilot. Finally, several application ontologies based on certain pilots will be developed to demonstrate how the ontologies are applied in real use cases. The structure of the QU4LITY Ontologies is as shown in Figure 4.

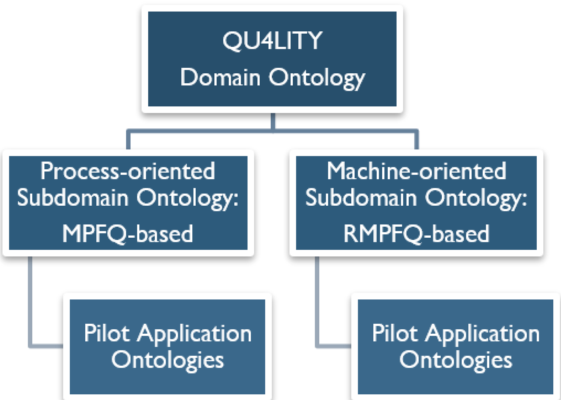

Figure 4 Structure of the QU4LITY Ontologies following the IOF Architecture.

3.2.2 User Story Mapping (USM) method

The User Story Mapping (USM) method [29] is adopted to define the vocabularies based on QU4LITY user stories by providing an efficient, time saving, bottom up requirements analysis for the design of Knowledge-Based Engineering (KBE) Tools. It is a user centric method which allows the designers of the software to learn what the future users expect from this KBE tool, as well as it helps the users to express their over-all demands in functional view which is close to them.

A user story map is a user centric approach and organizes the backlog along scenarios and users. It answers the question how a user uses the product. A bottom-up approach for the definition of domain concepts based on USM consists of the following five steps:

- Step 1: Apply the USM method, based on the business requirements and the project vision.
- Step 2: Gather other sources of information (standards, past experience...) in order to collect some generic and specific concepts, with respect to the scope resulted from the application of USM (Step 1).
- Step 3: Create a unique list of concepts that covers entire domain based on usages, roles and activities resulted through the application of USM and the generic concepts collected within the previous step.
- Step 4: Define relations and dependencies among the list of concepts.
- Step 5: Create a dynamic knowledge base covering the domain, expressed in some of the standard formats like relational data base, ontology, semantic model.

In the QU4LITY project, the input of WP2 task 2.5 includes the user stories and analysis of stakeholders' requirements. The USM methodology could be applied to the analyze results of certain pilots from relevant tasks and thus pave way to the development of semantic models. More details are introduced in the QU4LITY Vocabularies section.

3.2.3 Methods for designing data models

For the development of data models, we tried to reuse existing data models developed in previous research and projects. Many data models related to QU4LITY vision have been developed which provide solid basis for developing new data models. More details of existing models and approaches will be introduced in the QU4LITY data model section.

From the methodological point of view, we followed the critical action research method, in which the researcher and industrial practitioners collaborated in the analysis of the problem and in the development of a solution based on the analysis. Critical action research is based on the analysis, action, evaluation, and critical analysis of practices based on collected data, in order to introduce improvements in the relevant practices. This type of research is facilitated by the participation and collaboration of a number of individuals with a common purpose where the research focuses on specific situations and their context.

4. QU4LITY Vocabularies

4.1 User stories and stakeholders' requirements

Fully consideration of user stories and stakeholders' requirements is the precondition for developing data models and defining vocabularies. QU4LITY Task 2.1 "User Stories and Analysis of Stakeholders' Requirements" has analyzed user stories and stakeholders' requirements. It aims to collect, document and analyze requirements regarding the excellence in ZDM. The requirements will be collected and analyzed based on a variety of different modalities, including direct interactions with stakeholders (i.e. manufacturers and their employees, quality management experts, providers of ZDM solutions, digital manufacturing solutions integrator), focus groups, collection of questionnaire-based feedback, documentation of user stories reflecting the viewpoints of different users (e.g., maintenance workers & engineers, production quality managers), as well as review of relevant project's and initiatives (e.g., projects of the 4ZDM cluster, H2020 FoF-09 projects on maintenance etc.). Interactions with stakeholders will be facilitated by the partners' business networks, including clusters, associations (e.g., EFFRA) and DIHs where the consortium partners' play a leading role.

Some of the user stories are closely related to data modelling and vocabularies, which are listed as below:

- "As a data scientist I want a scalable platform that is fed by (near) real time data of current and new sensors of the entire shimmer production chain that describes the properties of the production process in such a detailed way that I can make prescriptive models that avoid any errors in the shimmer line."
- "As a data Scientist, I want to obtain insights provided by the data analytics through the development of algorithms fed by the furnace operational data, in order to facilitate the optimize the furnace operation."
- "As a data scientist I want a scalable platform that is fed by real time data from machine sensors of the entire automated line, including the CMM, containing all needed parameters for the deployment of efficient predictive and prescriptive algorithms, to be used for supporting the design, manufacturing and quality control phases of the production process according to the factory KPIs."
- "As an R&D engineer I want insights provided by analytics that help me understanding components that need to be improved."
- "As an innovation engineer, I want to gain physical insight in process steps provided by advanced sensing technologies combined with analytics that help me to understand error sources and to avoid them in machines and productions lines."
- "As a maintenance engineer I want a prescriptive functionality provided by analytics which indicates sufficiently in advance the time and component to be replaced so that maximum efficiency of the equipment is maintained, and defects are avoided on production parts."

- "As a production manager, I want an improved monitoring and data analysis platform in order to ease decision-making on the hot stamping line operation (mainly on the industrial furnace), enabling the optimization of the production process and so reduce (furnace) operational costs."
- "As Quality Test Engineer want to know what factors are affecting the quality parameters for each product and aggregated by model. This allows me to make decisions that can concern the set-up of the production areas as well as the programming of new test plans, aimed at solving product and process quality problems effectively and quickly. The information can be aggregated so that I can know what the factor is that most influences the reduction of quality: Material, Process or Function."

The survey of stakeholders' requirements shows that all pilots understand the need of communication and integration of Industry 4.0 components, which relies heavily on the communication standards. Although most pilots do not need to use all of the standards that have been anticipated, they are open to new standards if necessary. Especially, attention should be given to currently lacking or immature standards focusing Digital Data Models, i.e. ontology standards and standards regarding data models. The work done in pilots ensures positive steps towards the compliance to the QUALITY reference architecture, which is one of the essential requirements in the project. Most pilots are not using any knowledge management approaches like Ontology, Semantic Web. Only four pilots are using or developing Ontologies, which mainly focus on quality management and defect prediction. This leads to the requirements for data models and vocabularies:

- QU4LITY should extend/update the current list of standards to include the latest standards regarding interoperability for IIoT and digital twin applications, AAS discussions and integration of ontology, data models, and vocabulary,
- The compliance of the QU4LITY reference architecture must be verified against the latest standards, e.g. RAMI 4.0,
- QU4LITY should provide an upper-level Ontology and at least one domain Ontology with detailed specifications to showcase how such semantic modes can be developed and applied for knowledge management,
- The vocabulary for QU4LITY should be defined including the latest publications regarding terms and definitions, especially ZDM and quality aspects.

These user stories and stakeholders' requirements will be carefully considered during the development of data models and vocabularies to assure the output of this task aligns with application scenarios.

4.2 Competency questions for QU4LITY pilots

One of the most used methodology to design the taxonomies of industrial ontology is through Competency Questions. This methodology can be used to define the application domain boundaries and capture elements definition. By composing a top-level overview, abstract concepts facilitate to perform system architecture planning and optimization. After the extraction of entities from Competency Questions, the list of classes can be updated in comparison with existing ontology such as BFO, IOF, Product Lifecycle (PLC), and Product Service System (PSS) ontology. Thereafter, all the entities will be rearranged based on the BFO structure.

Competency question is one of the ways to determine the scope of the ontology and to sketch a list of questions to be answered [30]. These questions will serve as the litmus test later: Does the ontology contain enough information to answer these types of questions? Do the answers require a particular level of detail or representation of a particular area? These competency questions are just a sketch and do not need to be exhaustive [31]. Some exemplary competency questions extracted from some relevant projects are shown below.

- What machines/equipment are considered in each use cases?
- What stakeholders/actors are considered for each use case?
- What is BOM for each machine (as a hierarchy diagram)?
- What component is critical for asset management in a specific use case?
- What is the component scope?
- What are the failure modes for each critical component?
- What are the effects of each failure mode?
- What is the criticality of each failure mode?
- What kinds of actions are required before the failure?
- What kinds of actions are required after the failure?
- What kinds of sensor is available/required for each critical component?
- What kinds of critical components are linked to a sensor?
- What type of signals are collected from a sensor?
- Which sensor/s is/are relevant to detect a specific failure mode?
- Which is the signal unit of measurement?
- Which is the minimum value of the signal?
- Which is the maximum value of the signal?
- How often signals are stored in repository?
- Which is the sampling frequency of the signal?
- In which working phase is included the control system?
- In which working phase it is excluded the control system?
- Which kind of mathematical elaboration is requested for the signal? (average, standard deviation, root mean square, …)

Competency questions for each business scenario provide the fundamental principles for building the specific ontology. End-users' requirements are interpreted in the form of competency questions and the corresponding domain ontology will be designed by answering these questions. In the future, specific competency questions will be defined for different QU4LITY pilots based on the user stories and stakeholders' requirements.

4.3 Specification for vocabularies

Semantic interoperability is concerned with the packaging of data (syntax) as well as with the simultaneous transmission of the meaning with the data (semantics). This is accomplished by adding data about the data, called metadata, linking each data element to a controlled, shared vocabulary. The meaning of the data is transmitted with the data itself, in an information package that is independent of any information system. It is this shared vocabulary, and its associated links to an ontology, which provides the foundation and capability of machine interpretation, inferences, and logic.

Some efforts have been spent in previous projects on defining IOF top vocabularies. For example, based on the DOLCE foundational ontology [32] and UFO [33], top-20 IOF vocabularies have been characterized as follows [34]:

- Manufacturing resource
- Material resource
    - Process resource
- Assembly
- Component
    - Component part
- Product
- Product quality
    - Product feature
- Manufacturing machine
- Equipment
- Supplier
    - Provider
- Customer
- Plan
- Task
- Business process
- Manufacturing process
- Planned process
- Transport Process [Manufacturing]
- Design

The following detailed definitions for several vocabularies demonstrate how a vocabulary can be characterized [34].

- **Manufacturing resource**
  Collins dictionary:" The resources of an organization or person are the materials, money, and other things that they have and can use in order to function properly".
  Another, more compact definition is that resources are "available means".
  The fact that resources need to be in possession of the organization/person (i.e., available) is crucial.
  Note that the notion of resource is a relative one: something is a resource for somebody and for a purpose. The purpose may be generic. "
  Any entity which is under the control of an Organization and may participate to one of the Manufacturing Processes adopted by such Organization.
    - Examples:

a single screw present in the workshop (even as part of a product intended to be sold); the amount of metal that constitutes such screw, assuming it may be needed (even in melted form) in a process; a screwdriver present in the workshop;

- Counterexamples:
  a component which is needed but is not present in the workshop; a Product which is present in the workshop (not sold yet) but can't be used for a Manufacturing Process.
- Notes:
  An entity may participate to the process described by a Plan if it satisfies a Resource Description that is Part of such Plan.

These IOF top vocabularies will be the basis and references for defining QU4LITY pilots' ontology terms.

Properties with a predefined and standardized meaning are one pillar of the information exchange. Within the last years, domain-specific lists of properties have been standardized which allows an unambiguous characterization of products. Results of this comprehensive work done by the standardization committees within the last years are, for example, the international standard series IEC 61360CDD, IEC 61987, and the property library eCl@ss [35]. In these standards, thousands of properties of technical assets are semantically defined. Every standardized general property can be referenced by a global unique identifier. By this unique identifier as "semantic reference" and the name of the thing under consideration, two communication partners can agree that they mean the same property and can make statements to the value of this property. They need no semantic understanding what the meaning of the property is. These standards should also be complied when defining common vocabularies for QU4LITY semantic models.

Based on the user stories and stakeholders' requirements, the analysis of competency questions, and referring to the IOF top terms, the top terms for QU4LITY pilots are defined as listed in **Appendix B**.

5. QU4LITY data models

5.1 Reference data models

A series of data models have been developed focusing on different periods of product lifecycle. Several of them are closely related to QU4LITY scenario and could greatly facilitate the development of QU4LITY data models. Our future models will be based on these existing models.

- ### 5.1.1 MPFQ model

The MPFQ-model is named after its four main elements: Material (M), Production Process (P), Product Functions/Features (F), Product Quality (Q). This model was developed as part of the EU-project "inteGration of pRocess and quAlity Control using multiagent technologies (GRACE)" (GRACE consortium 2011). This model focuses on the manufacturing phase of the product while considering the strong interactions between product design and plant planning. At the manufacturing phase the planned product quality is brought into reality by assembling procured materials within production processes. The final product produced satisfies or dissatisfies the customer requirements and is being sold on the market. The roles of different elements of the model are illustrated in Figure 5[36].

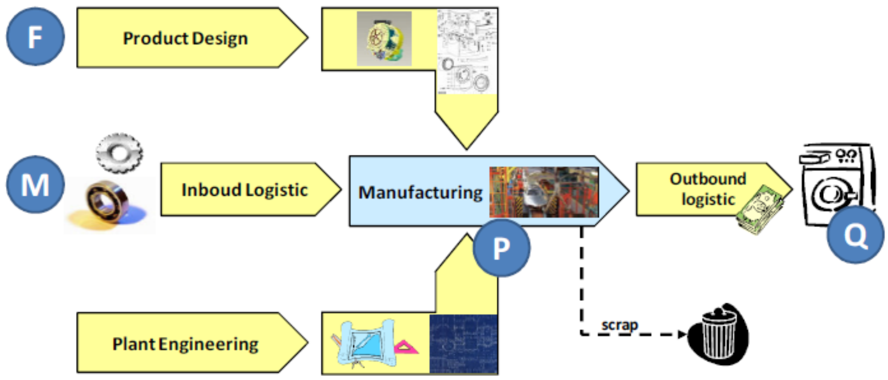

Figure 5 Central role of manufacturing for product quality.

The definition of the four elements of the MPFQ-model is [36]:

- **Material (M)** - as a collective term for everything that is needed to produce a certain product or product component. This may include raw materials (Oxford English Dictionary 2012b), pre-products, consumables (Oxford English Dictionary 2012a), operating supplies, product components and assemblies [37],
- **Production Processes (P)** - processing and transforming materials into the final goods by using machines, tools and human labor. This process is defined within the plant engineering. (DIN 8580),
- **Product Functions / Features (F)** - as distinguished characteristics of a product item. This is mostly focused on functionalities like specific tasks,

actions or processes the product is able to perform, but may also include other features like performance, etc,

- **Product Quality (Q)** - measured, following (DIN EN ISO 9000), as the degree of conformance of final product functions and features to customer requirements.

Figure 6 depicts the four main elements of the MPFQ-model and their interrelations. Within the MPFQ-model two types of interrelations can be found: the recursive dependencies between materials processes and functions and the straightforward dependencies of the MPF-part to the quality.

Starting with a manufacturing process (P) typically two or more materials (M) are combined to form a function (F) (see blue arrows in Figure 6). A Function is usually formed by one material acting on another (orange arrows in Figure 6). Hence, there is a closed interrelation loop given within the MPF-part. This fact is not surprising, as it shows the strong interlocking of plant engineering (combining materials (M) in production processes (P)) and product design (defining product functions (F) realized by materials (M)).

According to (DIN EN ISO 9000), product quality can be defined as conformance of product functions to customer requirements [38]. Thus, product quality is an aggregation of one or more product functions/features (black arrows in Figure 6). Despite these primary dependencies there are also processes and materials defining the product quality (black dashed line in Figure 6). This can be easily seen by taking the example of a green product footprint, which is e.g. depending on the energy consumption of the product and the resources spent during the manufacturing of the product. Thus, taking the example of a washing machine, the motor and heating element (materials) and the energy consumed during the single production processes primarily define the green footprint quality of the product.

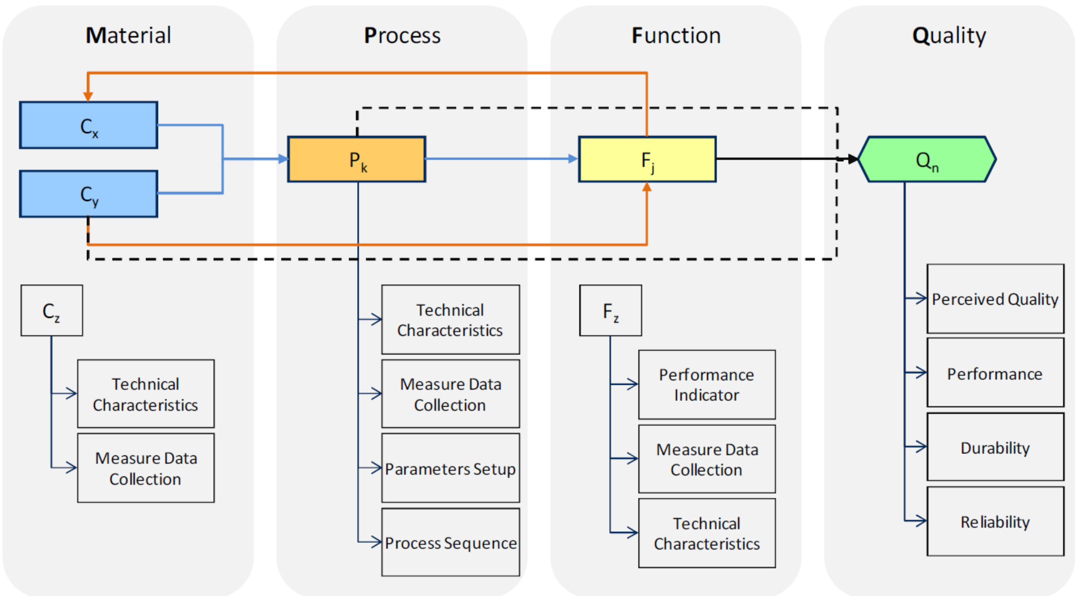

Figure 6 Four main elements of the MPFQ-model and their interrelations.

In order to address the requirements of adaptation, robustness and responsiveness, a multi-agent system (MAS) was developed in the GRACE project. The architecture of the MAS is composed of four types of agents [36]:

- **Product Type Agents (PTA)** represent the catalogue of products/parts that can be produced by the production line and contains the process and product knowledge required to produce the product, namely the product structure and the process plan. Note that PTAs not only act on plant/factory level but also on production line level,
- **Product Agents (PA)** handle the production of product instances along the production line (e.g., washing machines and drums). They possess a process plan to produce the product and interact with the agents responsible for the process and quality control,
- **Resource Agents (RA)** represent the physical resources of the production line, such as robots, quality control stations and operators. They manage the execution of their production/testing/transportation/assembly operations in the production line. The RAs comprise several specializations according to the particularities of the resource; namely Machine Agents (MA), Quality Control Agents (QCA), Transport Agents (TA) and Operator Agents (OA),
- **Independent Meta Agents (IMA)** implements global supervisory control and optimized planning and decision-making mechanisms, e.g. defining and adapting global policies for the system. In opposite to the PA and RA agents, that are placed at the operational execution level and are mandatory, the IMA agents are positioned in a higher strategic level and are not mandatory (the system can continue working without them, however losing some optimization).

In multi-agent systems the interaction among individual agents is crucial which requires a common understanding. The use of ontology provides a solution by formalizing the structure of knowledge including the concepts, the predicates (relations between the concepts), the terms (attributes of each concepts) and the meaning of each term (type of each attribute).

5.1.2 Z-BRE4K model

Z-BRE4K is a novel predictive maintenance platform to eliminate unexpected breakdowns and extend the life of production systems. A semantics-driven architecture for predictive maintenance has been developed by our group in this project, as shown in Figure 7.

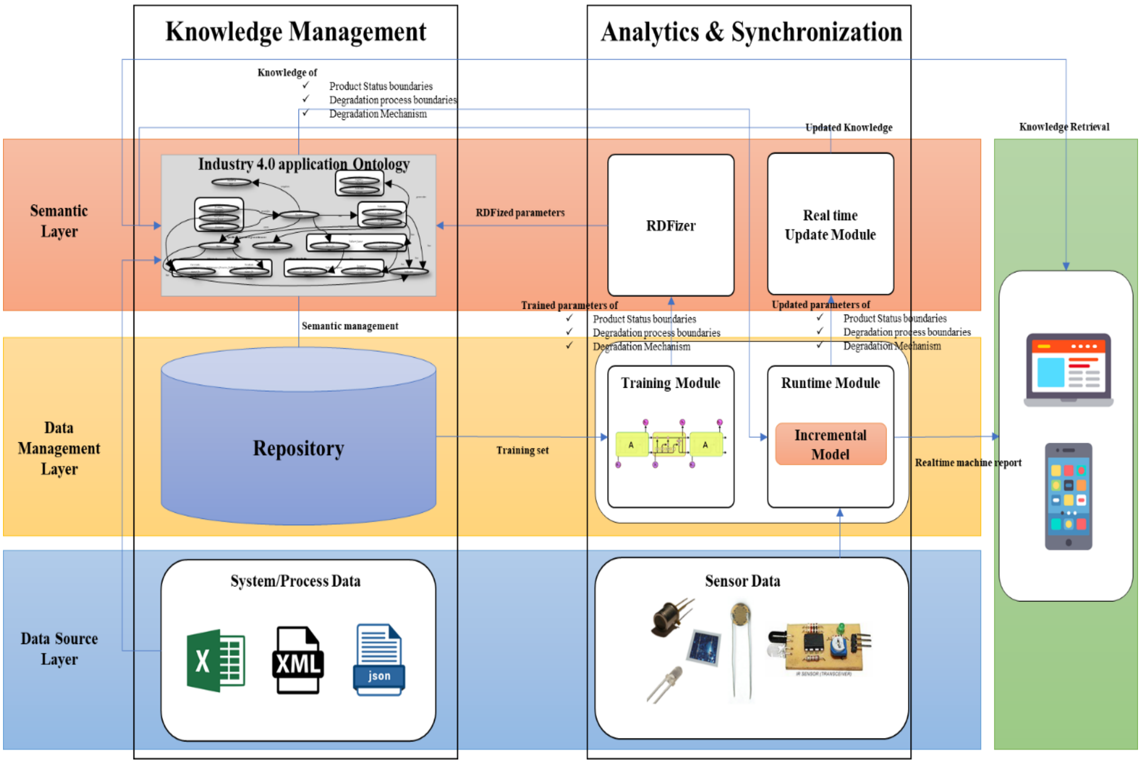

Figure 7 A semantics-driven architecture for predictive maintenance.

One of the key components of the semantics-driven architecture is the predictive maintenance semantic model, as shown in Figure 8.

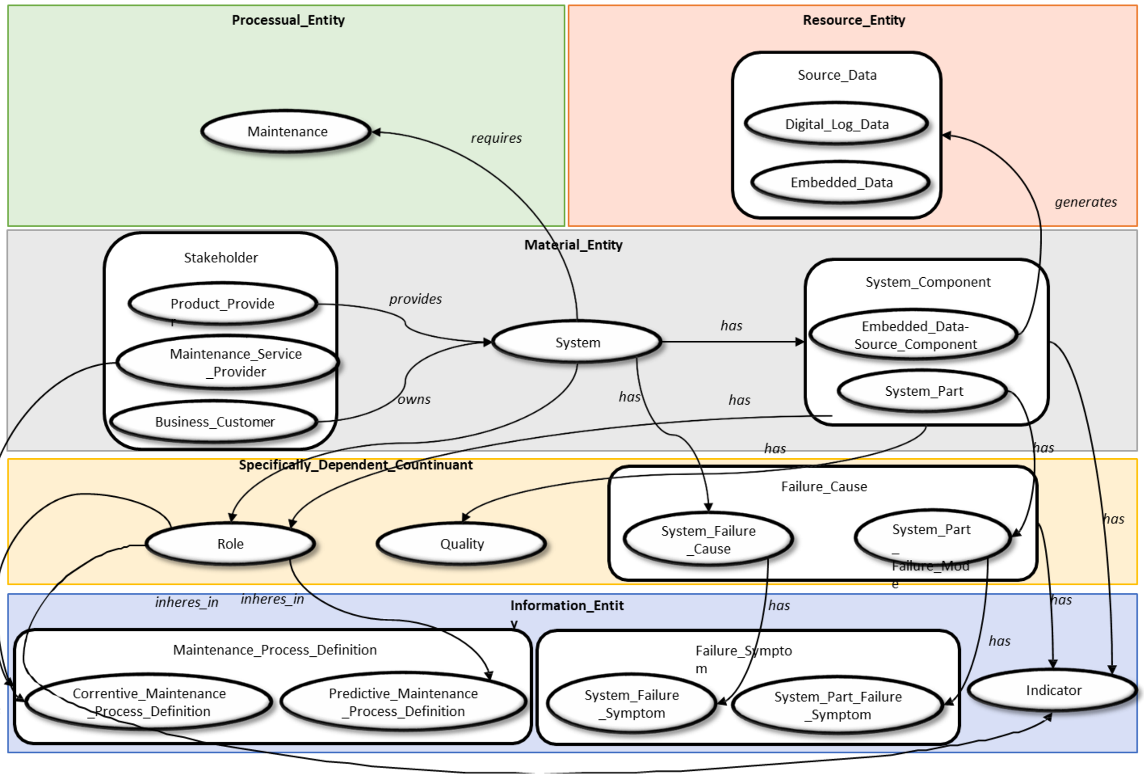

Figure 8 Predictive maintenance semantic model.

These data models, as mentioned above, are highly related to the requirements of QU4LITY project. With a deeper analysis it will be possible to extract reusable components and greatly benefit the development of data models for QU4LITY project.

5.2 QU4LITY data models

- ### 5.2.1 RMPFQ model

The MPFQ-model was developed based on the processes of assembly manufacturing. This model could properly describe an assembly process. However, when it is applied to a machining process, where usually only one workpiece is involved, and it interacts directly with a machine tool which might also affect the quality. Moreover, in many cases, the machining process has no direct impact on production functions. Therefore, the MPFQ-model needs to be improved to fit such machining processes.

To fit the scenario of machining processes, we adjusted the MPFQ-model by adding a Resource element to represent the manufacturing resources and changing the Function element to Function/Feature to represent the output of a machining process. Figure 9[1] shows the elements of the proposed RMPFQ-model and their interrelations, as well as some data related to these elements.

[1] A conference paper introducing this model has been accepted to be presented on the CIE50 conference (https://ejust.edu.eg/cie50/).

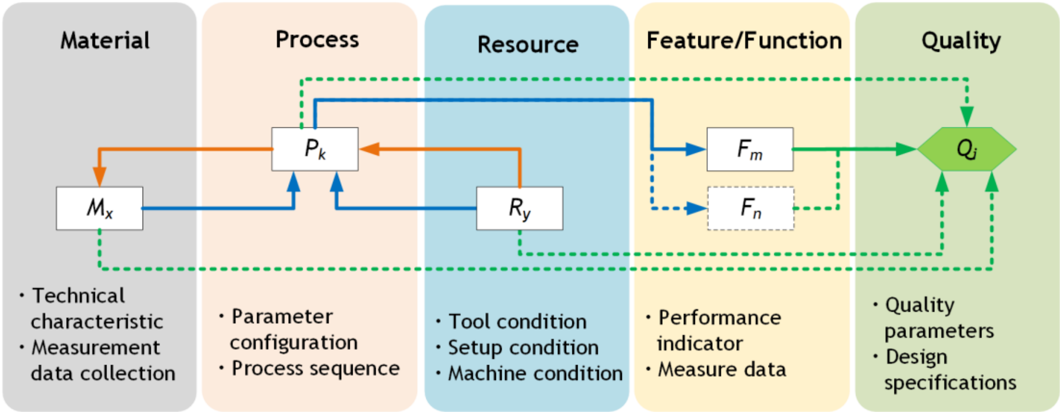

Figure 9 RMPFQ-model elements and their interrelations.

Based on the definition of MPFQ-model we define the elements of the proposed RMPFQ-model as follows:

- Manufacturing Resource, according to ISO 15531, represents the devices, tools and means, at the disposal of the enterprise to produce goods and services, but except raw material and final product components,
- Material represents everything that is needed to produce a certain product or product component, which may include raw materials, pre-products, consumables, operating supplies, product components and assemblies,
- Manufacturing Processes are defined as processing and transforming materials into the final goods by using machines, tools and human labour. This process is defined within the plant engineering,
- Product Functions / Features represent the distinguished characteristics of a product item, which may include functionalities like specific tasks, actions or processes that the product is able to perform; and/or other features like performance,
- Product Quality (Q) is defined as, according to DIN EN ISO 9000, the degree of conformance of final product functions and features to designed requirements.

For the definition of Resource, we followed the ISO 15531 standard as it is close to the purpose of the proposed RMPFQ-model. Personnel and material are not included as resource in the proposed model which is different from many definitions such as in IEC 62264. Another difference worth to notice is that in MPFQ-model the *F* was defined including both functions and features, although it focused mainly on function perspective for assembly manufacturing. In contrast, in the proposed RMPFQ-model, it focuses more on the feature perspective which is common for machining processes. Regarding the quality, it mainly refers to the conformance of product functions to customer requirements for assembly, while for machining processes, it mainly refer to the conformance of workpiece features to the designed requirements.

As shown in Figure 9, there are several types of interrelations among the elements of the RMPFQ-model. First, a given workpiece (M) is machined by machining resources (R), e.g. a given setup (fixturing and associated tooling) and a cutting tool (R), through a planned machining process (P), composing the RPM interactions (marked with orange lines). Second, the machining process (P) uses input material (M) and resources (R) to produce one or more features (F), composing the RPMF interactions (marked with blue lines). Moreover, all the RMPF-elements may also have straightforward impact on the quality (Q) of the machined workpiece (marked with green lines). There also exist relations among different resources, i.e. machine, setup, and cutting tool.

The RMPFQ-model is an extended version of the existing MPFQ-model, but not aims at replacing it. They focus on different phases of a product lifecycle, i.e. product part machining process and components assembly process, thus they are closely correlated with each other. In industrial practice, as shown in Figure 10, they are usually applied in different factories, i.e. RMPFQ in supplier manufacturers and MPFQ in assembly plants.

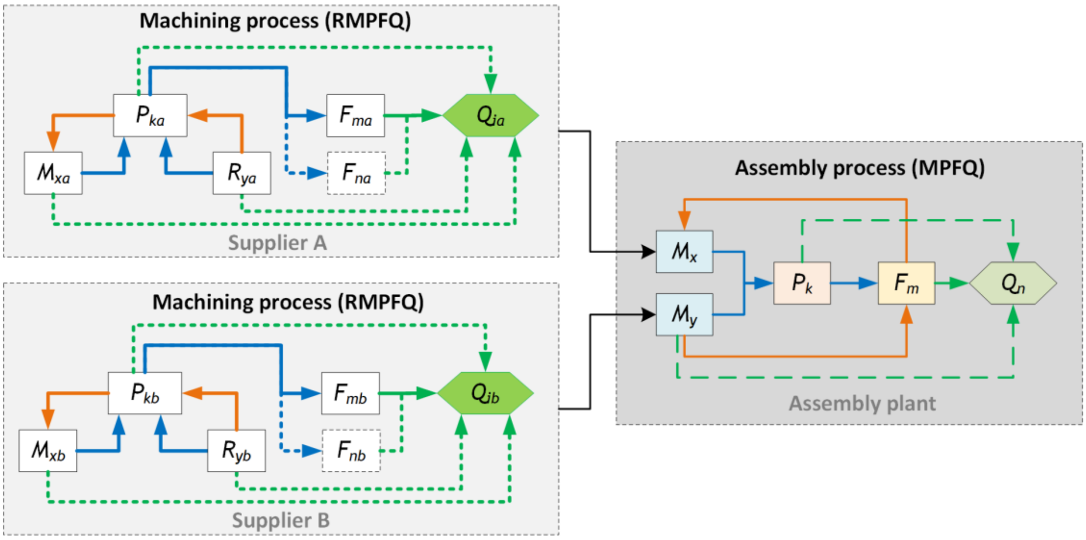

Figure 10 Correlations between MPFQ-model and RMPFQ-models

The elements of both models also have different emphasis although they might share the same names. The output of an assembly process, where MPFQ is applied, is usually either the final product or main components of it. In this case, the quality (Q) directly interacts with the customers' requirements, and the Function (F) is the core to fill such requirements, which then requires qualified components/parts/materials (M) being assembled through proper Processes (P). The configuration of the assembly process, e.g. setup, machine, fixture etc., is considered as part of the process. In contrast, the output of a machining process, where RMPFQ is applied, is usually a part manufactured by a supplier. The quality (Q) of the part does not necessarily interact with the customer's requirements, and the F element emphasizes more on the Features of the part although it may impact the Function of the product in the assembly process. The Resource (R) is separated from the Process (P) as they directly interact with the Material (M) and have much more impact on the quality of the machined part compared with the assembly process.

- ### 5.2.2 Semantic-driven digital twin model

A digital twin model should at least consist of three main elements, i.e. physical products in real space, virtual models in virtual space, and the connections of data and information that tie the virtual and real spaces together. Based on this three-dimension definition, a five-dimension DT model was proposed to promote the further applications of DT by adding two more dimensions, i.e. DT data and services, as shown in Figure 11. Semantic modelling has been used to improve data interoperability and cope with the data heterogeneous challenge. Domain ontologies were developed to capture and summarize intuitive information in a complex system using standardized languages. Augmented semantic capabilities can be added to digital twins by integrating semantic modelling technologies, thus to identify the dynamics of virtual model evolution and enhancing the decision-making capabilities.

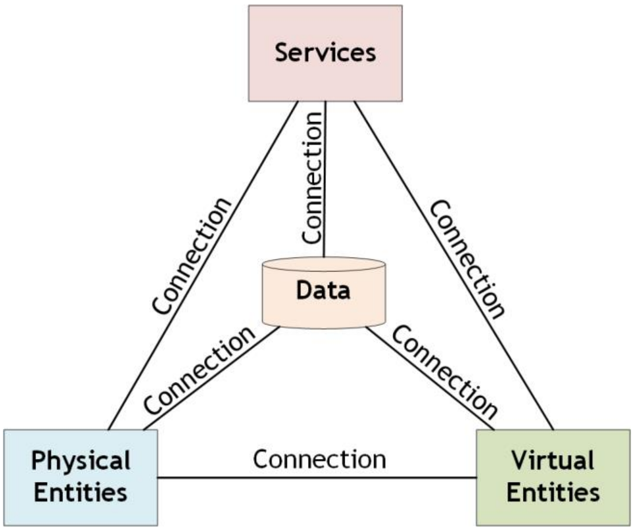

Figure 11 Five-dimension digital twin model

In order to facilitate the implementation of the proposed RMPFQ-model, a semantic-driven architecture is created by mapping the RMPFQ elements to the five-dimension digital twin model, as shown in Figure 12. Similar to the DT model shown in Figure 11, the proposed semantic driven model is also formed of five components, including the physical manufacturing system (Physical Entities), virtual models of the physical elements (Virtual Entities), data management based on semantic modelling (Data), Services, and the Connections among the four components. The details and functions of each component are introduced as follows.

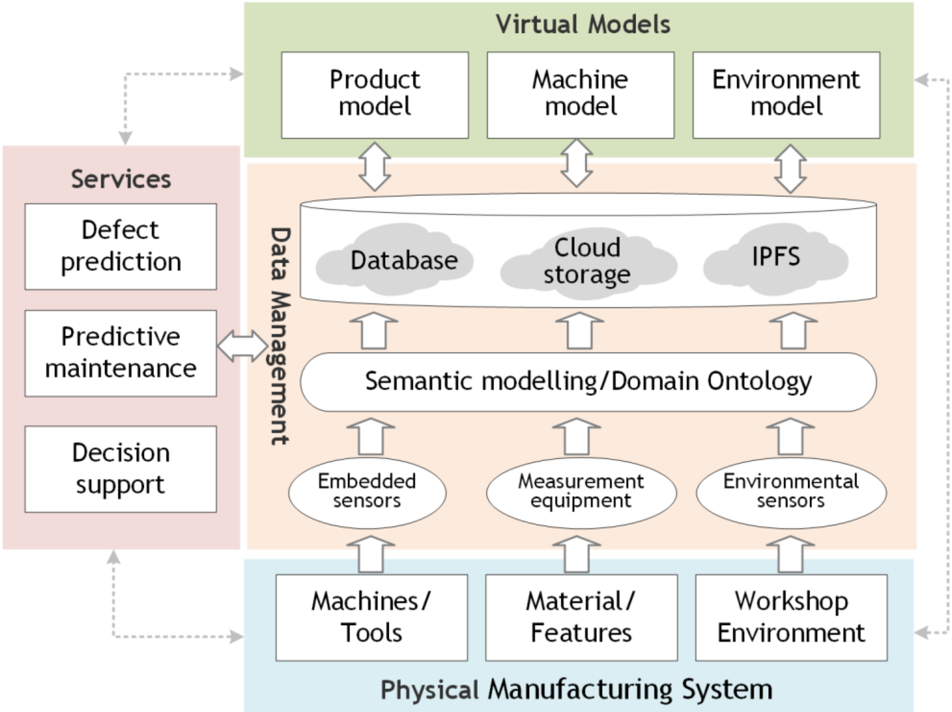

Figure 12 The proposed semantic-driven digital twin model for machining processes.

- Physical manufacturing system represents the physical elements that compose a manufacturing system, including resources (machines, cutting tools, setups), materials (raw materials, workpieces) and other relevant elements such as the environment of the workshop,
- Virtual models represent the digital representation of the physical elements, including the product design model, machine model and other relevant models such as the environment model. Depending on the strategy of a company, the machining process may be represented as a specific virtual model or as repository data in the data management component,
- Data management includes all functions related to data collection, integration, standardization, storage and analysis etc. Semantic modelling methods are used in this component to enable the interoperability among different data sources. Certain domain Ontology need to be created to summarize the common vocabulary definitions and the semantics of relevant knowledge in this domain. More details about this component is introduced in the following sections,
- Services represent the feedbacks from data analysis to both physical and virtual entities, such as production process optimization, fault diagnosis and prognosis, predictive maintenance etc. for physical entities; and calibration of the parameters of the virtual models during the running to sustain its high performance.

The data management component is the core of the proposed semantic-driven DT model as it connects all the other components. The data collected from the elements of the proposed RMPFQ-model are also integrated in this component. Details of each layer of the data management component are explained as follows.

- Data sources: According to the proposed RMPFQ-model, several types of data are involved corresponding to its elements. In a typical machining process, the R (machines, cutting tools, setups), M (material, workpieces) and F (features) data are usually generated on-line by embedded sensors or off-line by external measurement equipment. The environment data (temperature, humidity etc.) of the workshop are included in the proposed model considering they may indirectly impact the performance of the machining process and the output quality depending on the condition of the workshop, although it is not included in the RMPFQ-model. The P (machining process) and Q (quality requirements) data are usually obtained from data repositories of the production management systems like Product Data Management (PDM) or Enterprise Resource Planning (ERP) systems, which can eb mapped to the data storage layer as introduced below.

- Semantic modelling enables semantic interoperability among heterogeneous data sources. Ontologies provide formal definitions for the vocabulary and enable their logical capabilities, like reasoning and autonomous decision-making, by formalizing the structure of the knowledge such as the entities, the relations between entities and the attributes of each entity etc. The development of domain ontology for machining processes is based on the proposed RMPFQ-model which provides fundamental elements related to product quality. Technically, the upper-level ontology, like the Basic Formal Ontology (BFO), and existing initiatives, like Industrial Ontologies Foundry (IOF), should be followed to ensure interoperability among domain ontologies. In terms of development tools, the open source ontology editor Protégé can be used to edit and validate the ontology schema. It supports Web Ontology Language (OWL) and includes deductive classifiers to validate that models. It has an easy connection with agent development frameworks such as Java Agent Development Framework (JADE).
- Data storage and sharing: Depending on the demand of the company, different data storage and sharing solutions can be utilized, such as centralized enterprise data server, private or public cloud storage and decentralized distributed file system etc. Considering the data heterogeneity of a modern company, the system needs to support both structural and non-structural data storage and sharing. There is a trend adopt blockchain and distributed ledger technology to cope with data privacy/security and trust issues during data sharing, especially for inter-enterprise interoperations. To produce the information for desired services, data analysis needs to be conducted with the support of data mining and machine learning technologies.

Semantic models enable to capture complex systems in an intuitive fashion, which can be summarized in standardized ontology languages, and come with a wide range of off-the-shelf systems to design, maintain, query, and navigate semantic models. Knowledge Graph (KG) is a more advanced semantic technology. It can acquire and integrate information into an ontology and utilize a reasoner to derive new knowledge and they can model information in the form of entities and relationships between them. KGs can accelerate the implementation of digital twins and they are considered as one of the main enabling technologies for the next generation digital twins to link and retrieve all kinds of data, descriptive and simulation models etc. Graph-based query languages enables to extract and infer knowledge from large scale production line data, to help generate digital twin models and therefore enhance manufacturing process management with reasoning capabilities.

With the help semantic models, heterogeneous digital twin models across the full lifecycle of a system can be integrated. It paves way for the next evolution of digital twins with augmented semantic capabilities for identifying the dynamics of virtual model evolution, promoting the understanding of interrelationships between virtual models and enhancing the decision-making based on digital twin. Figure 13 shows a conceptual example of such semantic-enhanced digital twin model. The ontology of virtual models describes across-domain models, with fact that it also identifies their interrelationships. The physical entity is defined as an aero-engine, the virtual entities may include CAD models, performance models, information models, FEM models, and CFD models etc. These models correspond to different phases of the aero-engine's lifecycle. The ontology is developed as core to formalize interrelationships between all the areo-engine models. This concept serves as one of the scientific innovations of this project and paves way for future development in this domain.

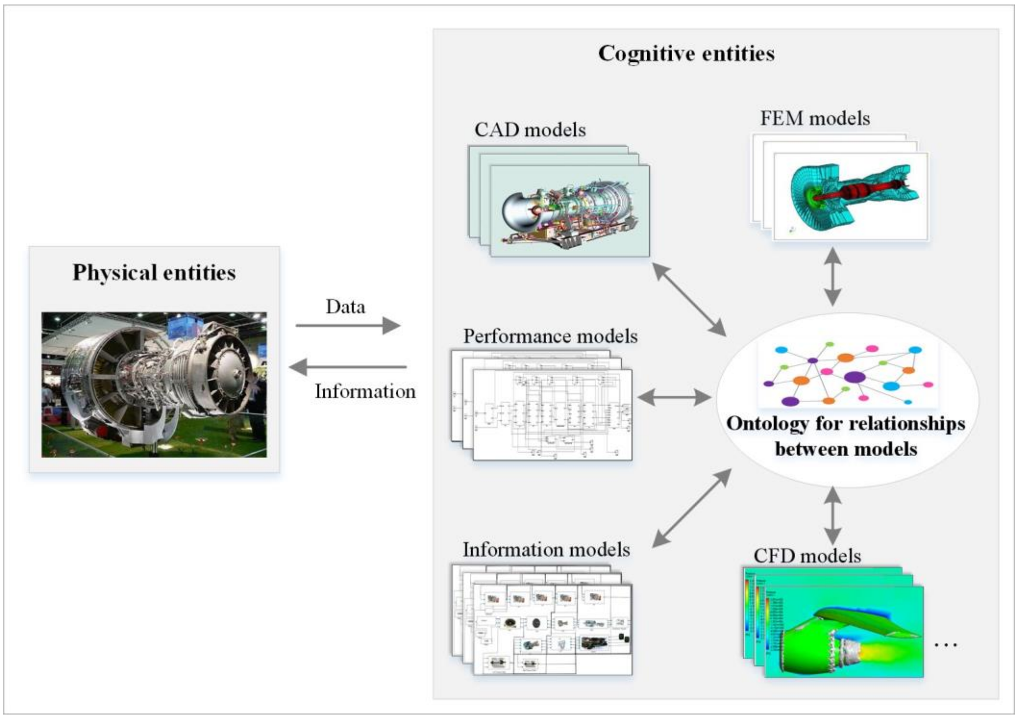

Figure 13 Semantic-enhanced digital twin model concept and its main elements i.e., physical entities, virtual entities (including multiple virtual models and Ontology models) and the communication between them [39].

6. Semantic modelling and Ontologies

6.1 Top-level ontology

A top-level ontology contains very general terms including "object", "property", "relation" etc., that are common across all domains. It can support broad semantic interoperability among a large number of domain-specific ontologies by providing a common starting point for the formulation of definitions. Terms in the domain ontology are ranked under the terms in the top-level ontology, e.g., the upper ontology classes are super-classes or super-sets of all the classes in the domain ontologies. For QU4LITY ontologies, the IOF ontology which has been introduced in previous sections, is used as top-level ontology. IOF is an evolving initiative which takes BFO and CCO as foundations.

IOF provides the IOF-Core ontology[2] which contains top terms that can be used as starting point for creating domain ontologies and application ontologies for QU4LITY. The main classes of the IOF-Core ontology are shown in Figure 14. The definitions and properties of the classes are available in the core ontology which is open access following the above link. According to the AQ vision and user stories and stakeholders' requirement, lower-level classes are added to the IOF-Core ontology to create domain-level and application ontologies. The general structure of the ontologies is introduced in the methodology section. In the following sections, the details of these ontologies corresponding to different levels will be introduced.

[2] https://github.com/NCOR-US/IOF-BFO/tree/IOF-Core-2020

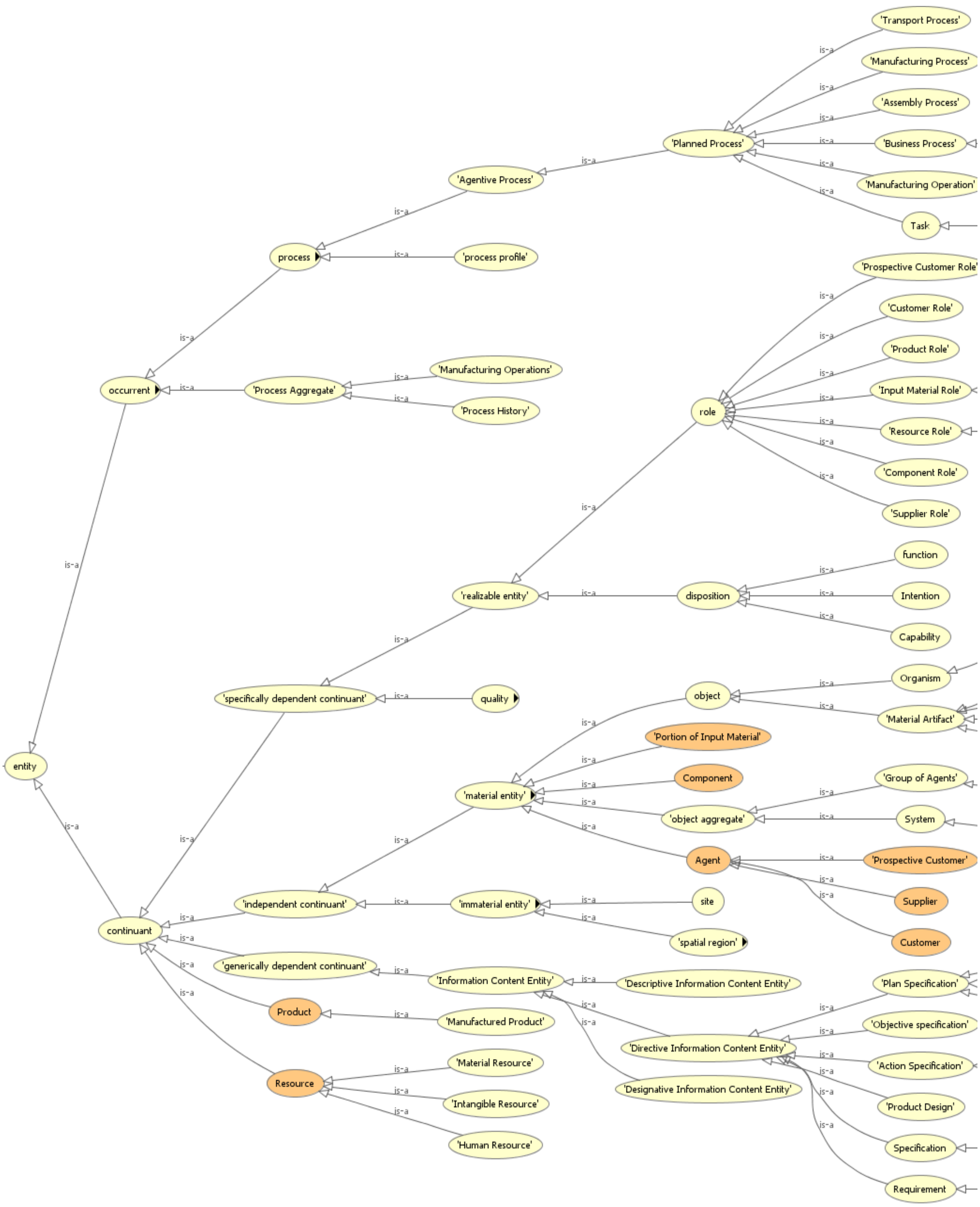

Figure 14 IOF-Core ontology classes.

6.2 QU4LITY domain ontology

One of the principles when developing the QU4LITY domain ontology is following the IOF top-level ontology. The IOF-Core ontology has defined many common terms suitable for most industry domains. For the QU4LITY domain ontology we try to reuse these terms as much as possible and add extra terms when no suitable ones available but needed according to the user stories and stakeholders' requirements, as well as the QU4LITY AQ vision. To formalize the development process, we mapped the RMPFQ model to the IOF-Core ontology and create new classes when an element is not defined. The main classes and the corresponding to the RMPFQ model are shown in Figure 15. The complete ontology is publicly available on Webprotégé website[3]. It is worthy to notice that both the IOF-Core ontology and the application scenarios are evolving frequently. Therefore, this QU4LITY domain ontology may also be updated in future accordingly.

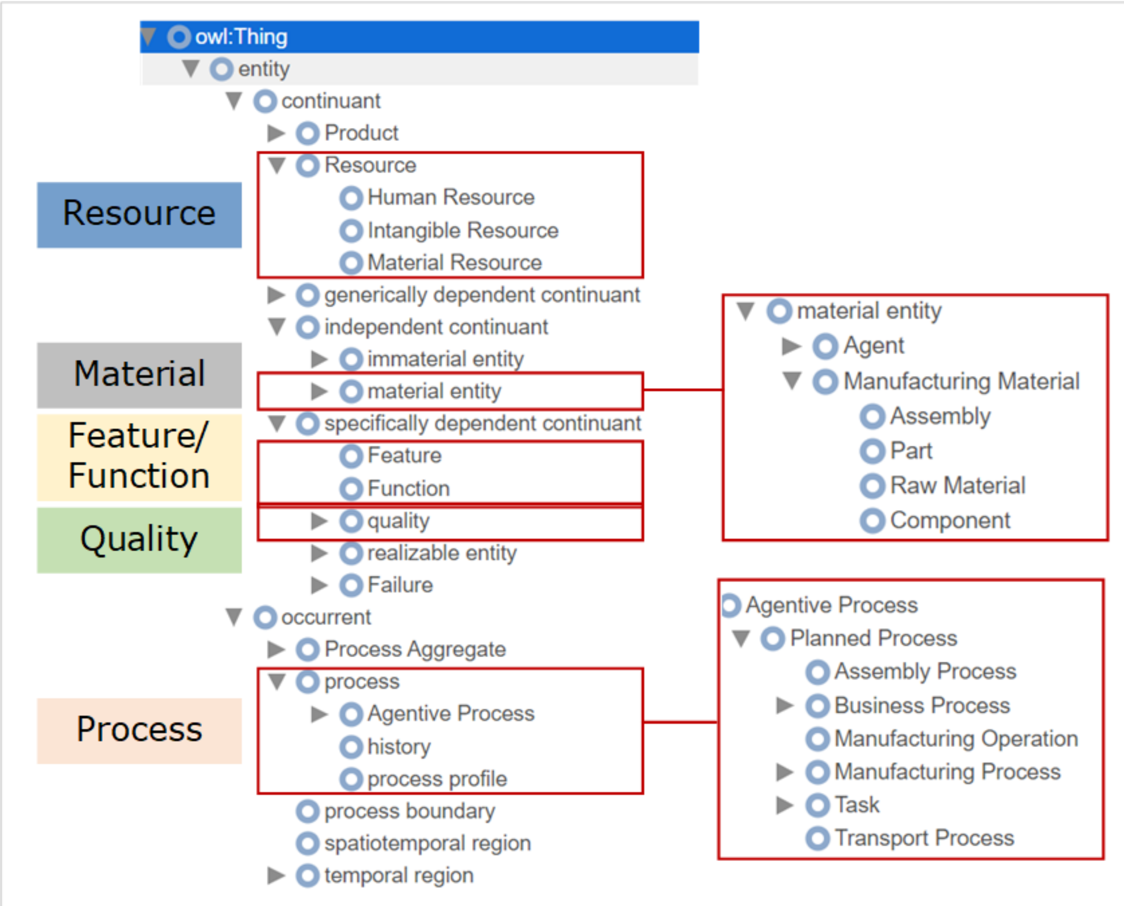

Figure 15 Structure and main classes of QU4LITY domain ontology.

6.3 Application ontologies

Application-level ontology is the lowest ontological level which aims to represent specific application cases with highly specialized classes and individuals such as a device from a specific manufacturer, a work station, a production line etc. The application ontology will often use or reference domain ontologies to construct

[3] https://webprotege.stanford.edu/#projects/8b81796d-6c0c-4cfb-875a-df68a444c1af/edit/Classes

ontological classes and relationships between classes. For QU4LITY pilots, application ontologies can be developed follow the QU4LITY domain ontology introduced above.

In order to demonstrate how application ontologies are developed and used in real cases, we choose two pilots from the 14 pilots of QU4LITY project as examples. The Airbus pilots is selected to represent the process-oriented scenarios, and the GF pilot is used to represent the machine-oriented scenarios. Similar application ontologies can also be developed following the same approach for other pilots when necessary.

- ### 6.3.1 Airbus pilot application ontology

  - #### *Application scenario*

The Airbus pilot aims to create a trade space framework for autonomous quality manufacturing systems' design. It focuses on the R&D phase of the assembly line for a new model of aircraft. During the early phase of an aircraft program, industrial architects are evaluating different industrial scenarios. One of the activities is to perform trade-off to optimize the future industrial architecture using different performance parameters like labor cost, industrial assets cost, lead time and different kind of defect risks within the extended enterprise. To cope with these challenges, product line and co-design concepts needs to be tackled. An MBSE approach where the Industrial System is seen as a System like any other (in parallel of the Aircraft) will allow to structure and to optimize the development of Industrial System and will allow to perform trade-off in a more efficient manner.

The proposed scenario is one assembly process: the fuselage orbital junction process to be designed for one station (Station 40) of the Final Assembly Line (FAL) for the Aircraft Product A321. The trade-off to be made is between a manual process and an automated process using a flex-track robotic mechanism, to achieve defined industrial performance requirements. The build process is as shown in Figure 16.

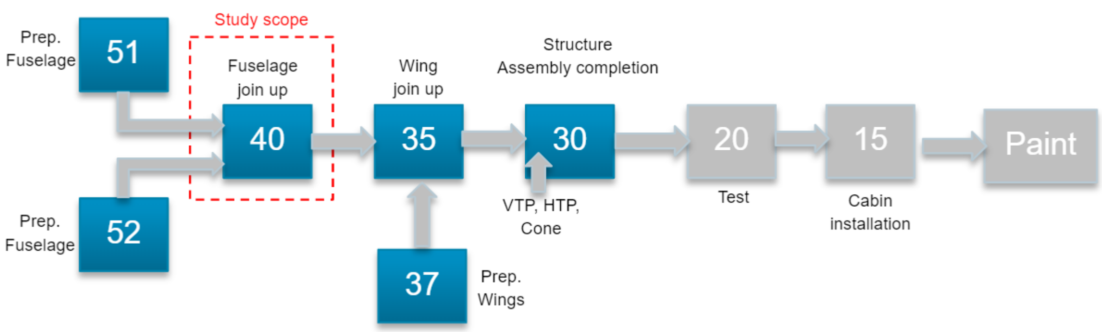

Figure 16 Assembly process of the aircraft.

The trade-off is expected to be performed between Manual Process and Flex Track process. The main differences between them are:

Manual Process:

- External drilling operations are performed by the operator (semi-automatic process),
- Internal drilling operations are performed by the operator (semi-automatic process).

Flex Track process:

- External drilling operations are performed by the Flex Track,
- Internal drilling operations are performed by operator.

The as-is industrial or manufacturing system design is based on MBE methods, for both product industrialization (by Manufacturing Engineers) and industrial system design (by Industrial Architects), similar to the ones shown in "next section/4. Behavior models". No SE or MBSE methods are applied.

Generally, Industrial requirements are managed on tables with manual traceability during process and resources design process. Trades are done to select different process and resources design options, analyzing disconnected simulations. Decisions are made based on Manufacturing Engineers and Industrial Architects experience during a collaborative engineering process. In order to assure the operational performance of the industrial system in operations, as per defined program requirements, an MBSE method is being adopted to support the Industrial system design, with a current TLR7 maturity level as shown in Figure 17.

Figure 17 Airbus pilot industrial system design process.

According to the application scenario, the functional architecture for this pilot is defined. A shown in Figure 18, it contains several function blocks including Requirement Management block, Architecture Definition block, Visualization block, System Integration block, Verification block and Simulation block. Semantic modelling plays an important role among these function blocks. It is the core of the System Integration block which integrates all relevant data and information from other blocks.

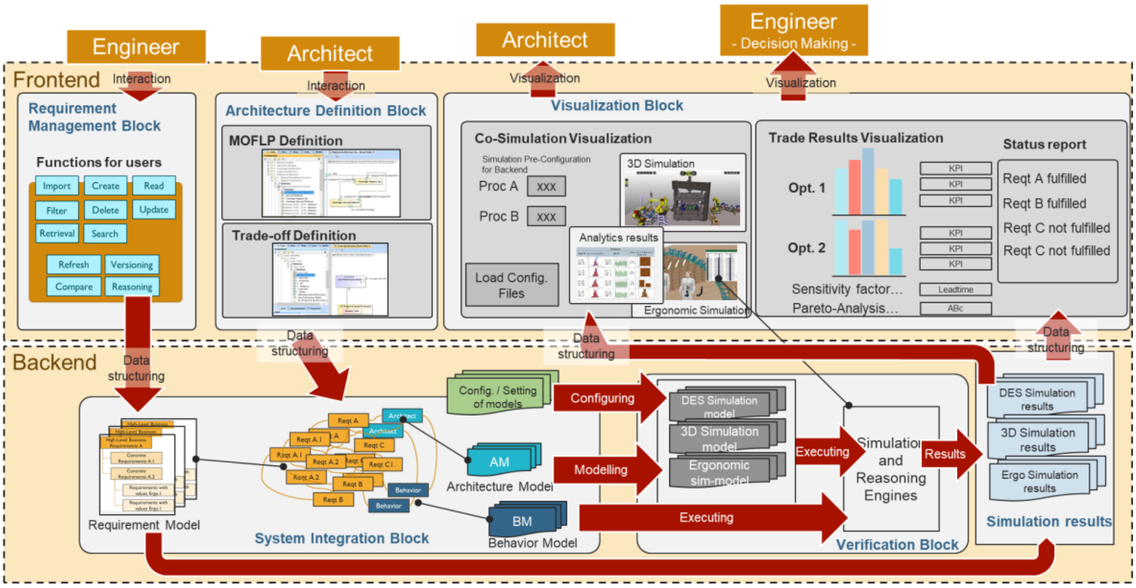

Figure 18 Functional Architecture of Airbus pilot.

- ***Airbus application ontology***

The application ontology of Airbus pilot aims to integrate information about system requirements and behavior models, and then provide support for simulation, as shown in Figure 19. The data flow of the pilot is shown in Figure 20. As the first step, both ontology and behavior model (SysML model) will take process diagram as input to assure the alignment. The ontology will provide input for simulation (in JSON or XML). Once the knowledge base is created, in future phases, the ontology will be updated automatically according to the behavior models.

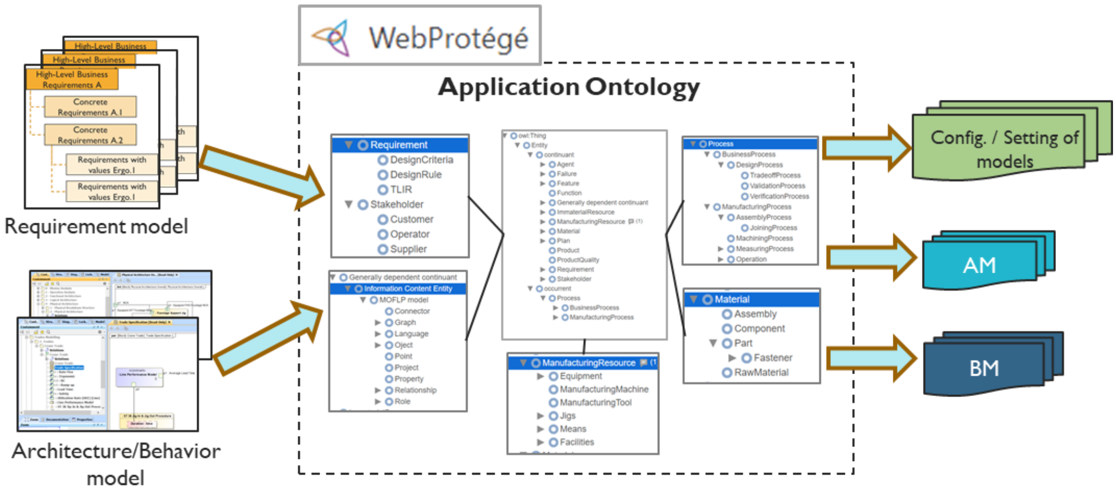

Figure 19 Airbus application ontology and connections with other function blocks.

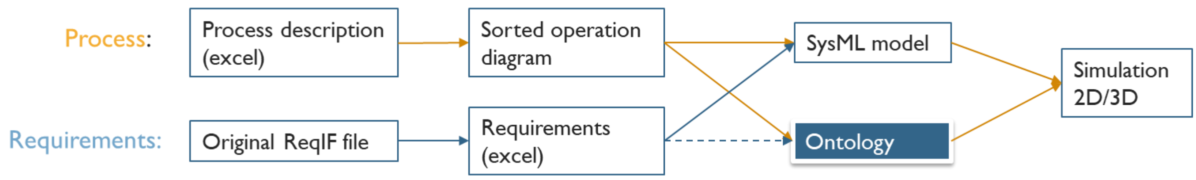

Figure 20 Airbus pilot dataflow.

The complete structure of the Airbus application ontology is shown in Figure 24. The original ontology is open available on Webprotégé website[4]. According to the application scenario, relevant ontology individuals are created corresponding to different operations, to model the domain knowledge of the assembly process. Some of the individuals are shown in Figure 21. In order to create connections between SysML behavior models and the assembly domain knowledge, some SysML model entities are also added to the application ontology, including both classes and individuals. Some of the individuals are as shown in Figure 22.

As future actions, once the dataflow is verified, the SysML and Ontology developed in the previous steps can be used as first knowledge to support new process development. Certain platform/DB can be added to manage the SysML and Ontology output to support 2D/3D simulation and process design, as shown in Figure 23.

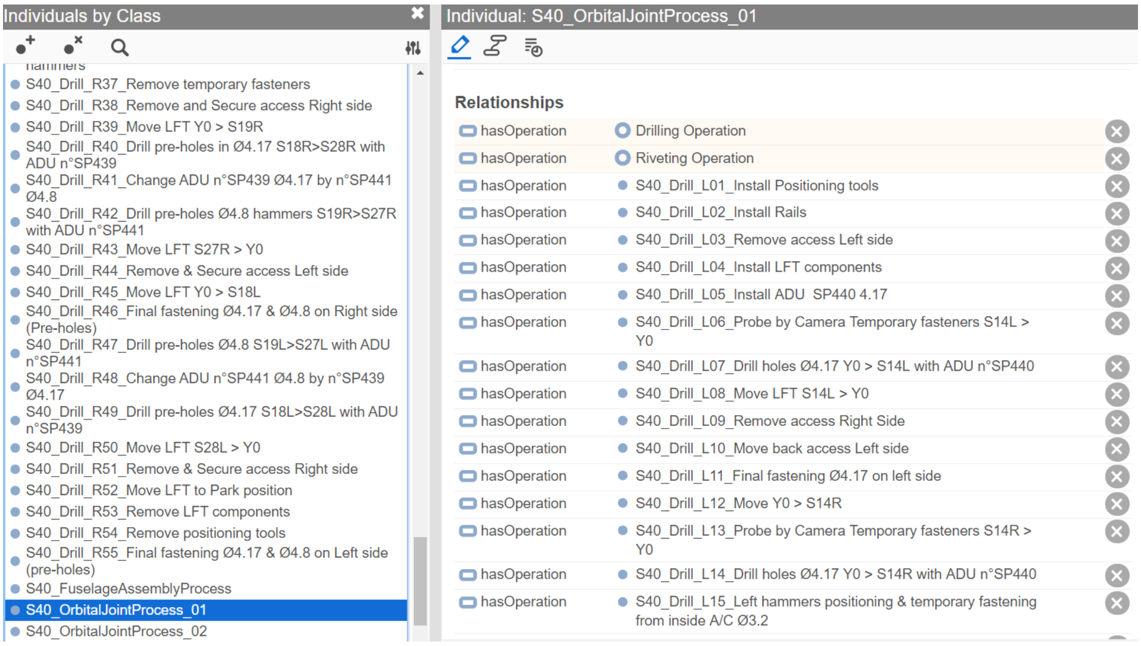

Figure 21 Airbus ontology individuals corresponding to fuselage assembly operations.

[4] https://webprotege.stanford.edu/#projects/cb52a4bf-acff-43e8-957e-9596a069a42d/edit/Classes

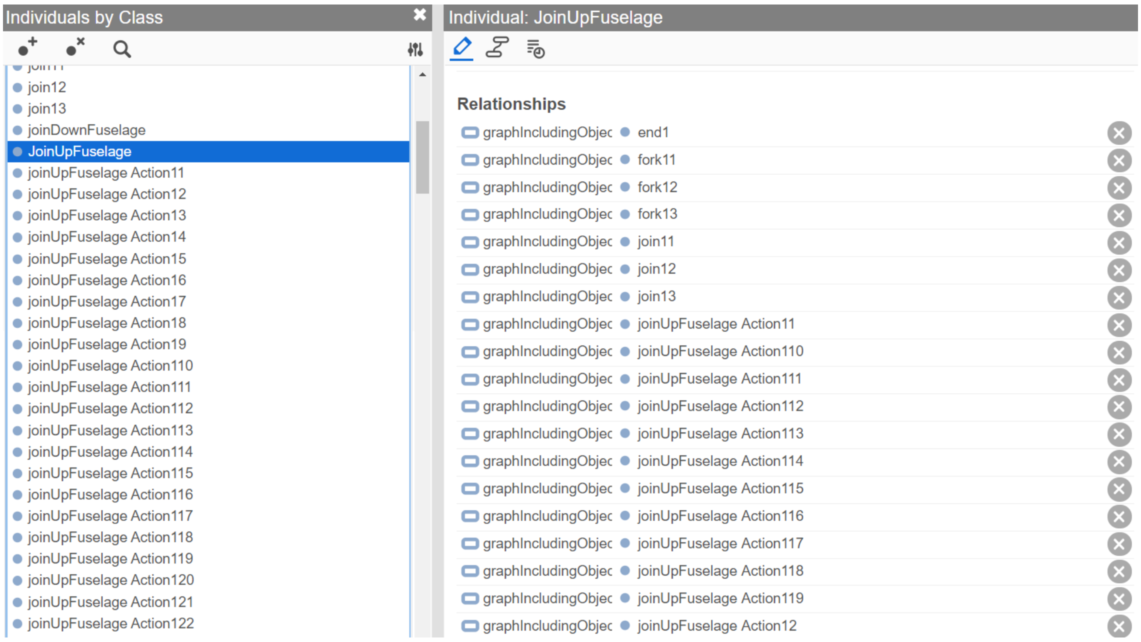

Figure 22 SysML model individuals of the Airbus ontology to connect SysML models with domain knowledge.

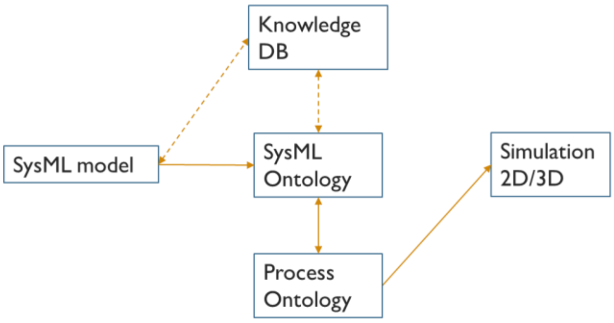

Figure 23 Final workflow for semantic-driven assembly process design and optimization.

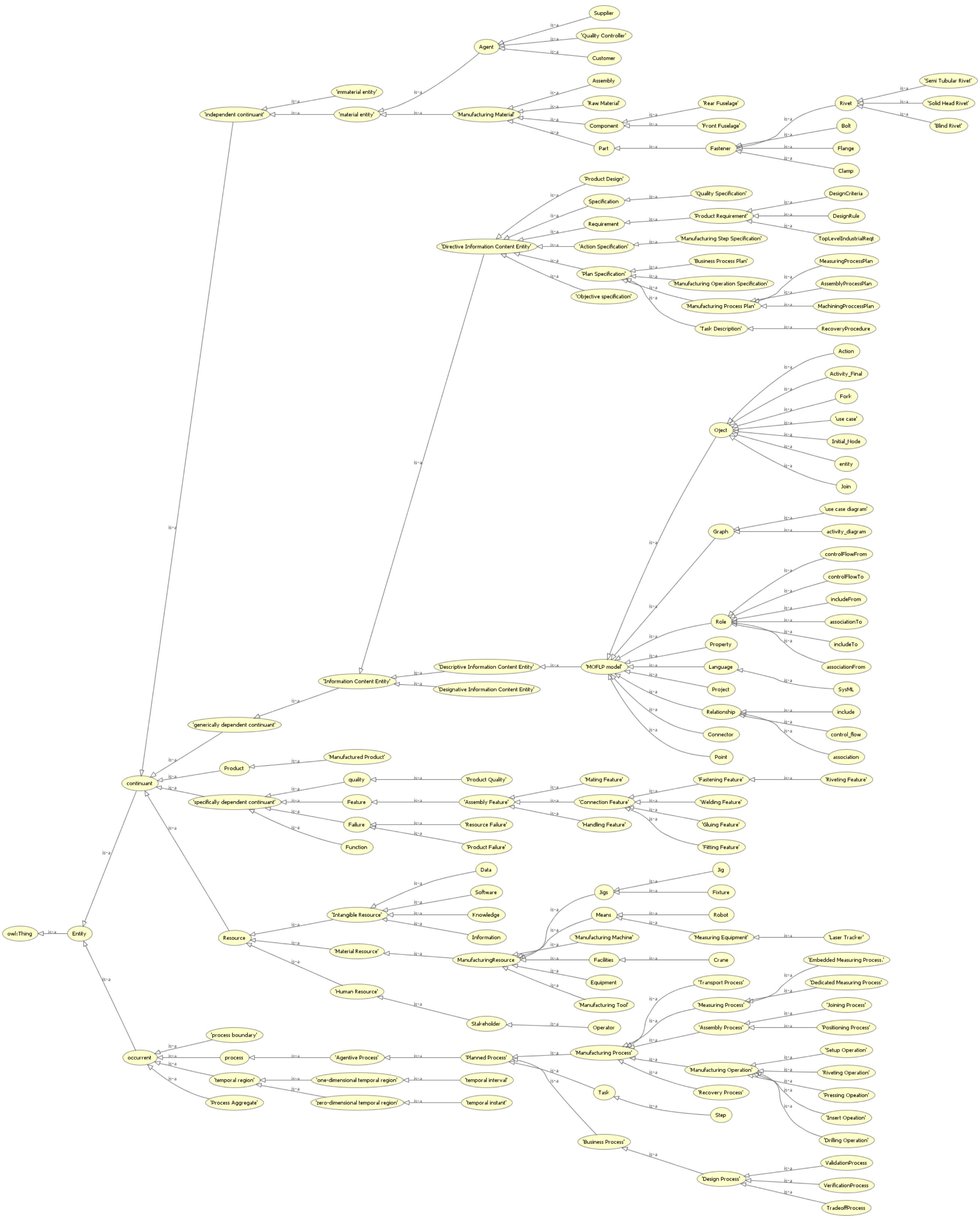

Figure 24 Airbus application ontology structure and entities (zoom in to check details).

• 6.3.2 GF pilot application ontology

▪ *Application scenario*

The current automated solutions integrate GF milling, measurement systems and automation systems, including Fanuc robots and dedicated GF software. The current systems require however extensive tests before operations and demand intensive maintenance in order to sustain productivity, which currently increases costs and presents high barriers for market development in the aerospace segments. Variances in quality are difficult to eliminate due to the effect of hardware and mechanics aging or defaulting on the effectiveness of predictive models used for tuning machining processes, this would require access to data from machinery, process and dimensional/surface quality measurements, currently incompatible and non-standard. Integration of data and information provided by different hardware and software suppliers working together in the shop floor, requires cost-intensive human intervention for aggregation and synthesis.

These main concerns of this pilot are as follows; i) the acquisition and storage of data, ii) data are spread across the product and factory lifecycles, iii) data is highly heterogeneous, unstructured and hard to analyze and distribute, iv) data analytics on operating data and monitoring, v) continuous evaluation and prediction of the health status of the equipment as cyberphysical system (i.e. transform descriptors extracted previously in relevant behavior models, allowing to represent the ways of functioning of the machine and the evolution of the equipment condition over time for detection and prognosis of failures), vi) decision-making support by considering the context of use of the equipment. To overcome these concerns, the main approach for GFMS pilot is inspired by exploitation of advanced enabling technologies for monitoring complex equipment. It requires integration capacity of heterogeneous data sources. In addition, its innovation lies on incorporation of various domains of knowledge.

Ontology facilitates a multidisciplinary approach through representation of domain knowledge with its concrete definition and provides semantic interoperability. Ontology, as a main reference model, provides a common glossary to integrate various data sources so that it facilitates covering our life cycles of a factory and products. In addition, it works as a meta-model to standardize the integration of further data sources. To recognize semantic context of data brings values as follows: i) End users can easily recognize and identify the meaning of individual data, and search meaningful data, ii) Software developers can harmonize data from various sources and request required data for each system components, and facilitate design of machine understandable entities on an intelligent engine, and iii) Requirements of end users can be satisfied since knowledge representation acts as a bridge between end users of a software platform and platform developers for design to meet requirements of end-users.

On the other hand, data analytics tools will exploit +GF+ data to monitor statuses of machines, and based on these results, it will support decision-making of optimization of maintenance policies. Currently, data does not have annotations describing machine status or maintenance history, even if an amount of data is available. In other words, the company has limited options to analyze manufacturing data, even if advanced machine learning techniques support to find the criteria or failure symptoms and optimize the maintenance schedule. Accordingly, the research should deal with the method of how to make a bridge between the current state of data and future maintenance annotation recorded in a shop floor. The general semantic-driven application framework is shown in Figure 25.

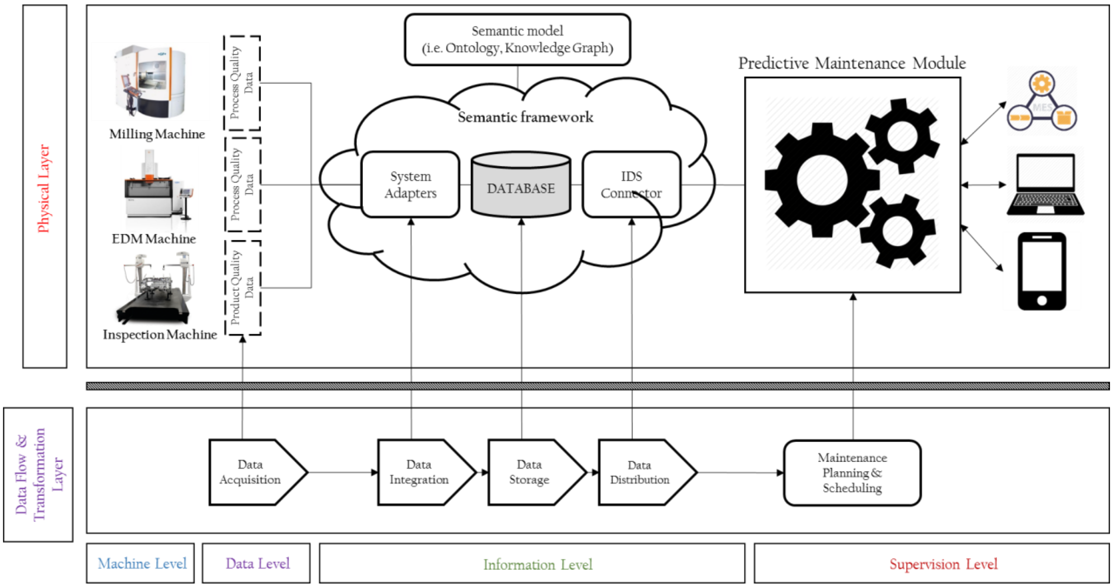

Figure 25 The general semantic-driven application framework of GF pilot.

More detailly, this pilot aims to implement the defect detection and predictive maintenance strategy bases on the case of machining an aeronautical turbine combustion case, as shown in Figure 26.

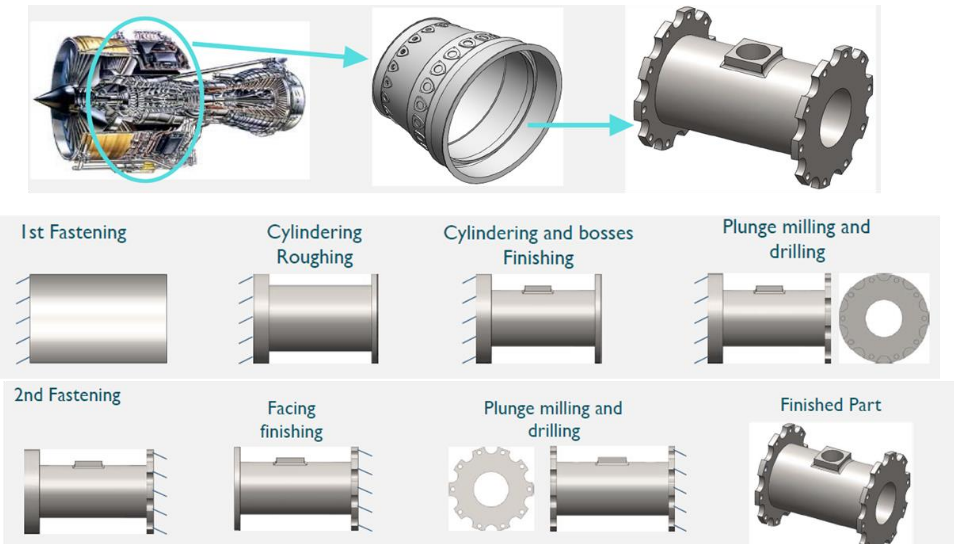

Figure 26 Combustion case and its machining processes for the GF pilot.

- ***GF machining application ontology***

The GF pilot application ontology follows the machine-oriented subdomain ontology. It aims to formalize the domain knowledge of the machining process to provide support for defect detection, predictive maintenance and process optimization etc. The domain experts including the machining engineers, machine manufacturers, process planning experts etc., work together to define the machining process, operation parameters, possible defects and data collection approaches etc.

Based on the domain knowledge provided by experts, an application ontology is developed. Some of the main classes and their relations are shown in Figure 27. The complete ontology is shown in Figure 29 and the original ontology is open available on Webprotégé website[5].

To facilitate the implementation of the ontology, the individuals corresponding to the machining operations are added to the ontology. Some of the individuals are shown in Figure 28. This pilot is currently under implementation and more individuals will be added and updated with the progress of the pilot.

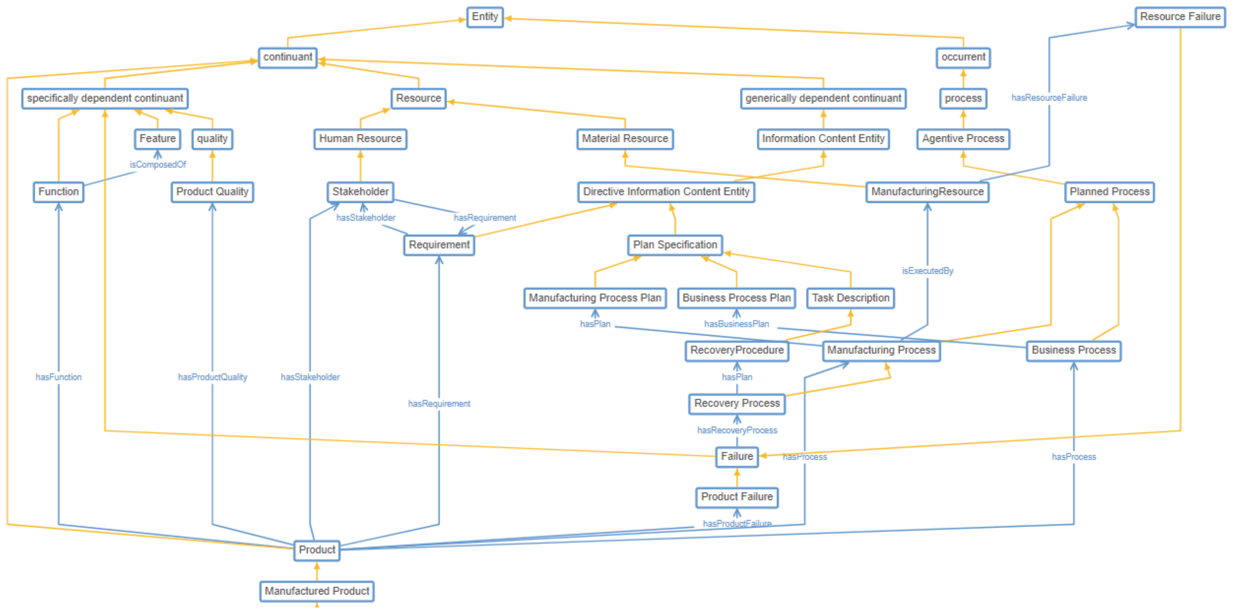

Figure 27 GF pilot application ontology classes and their relations (zoom in to check details).

[5] https://webprotege.stanford.edu/#projects/6fdf2f03-9f3e-413a-b06d-a9a81f1ec5f0/edit/Classes

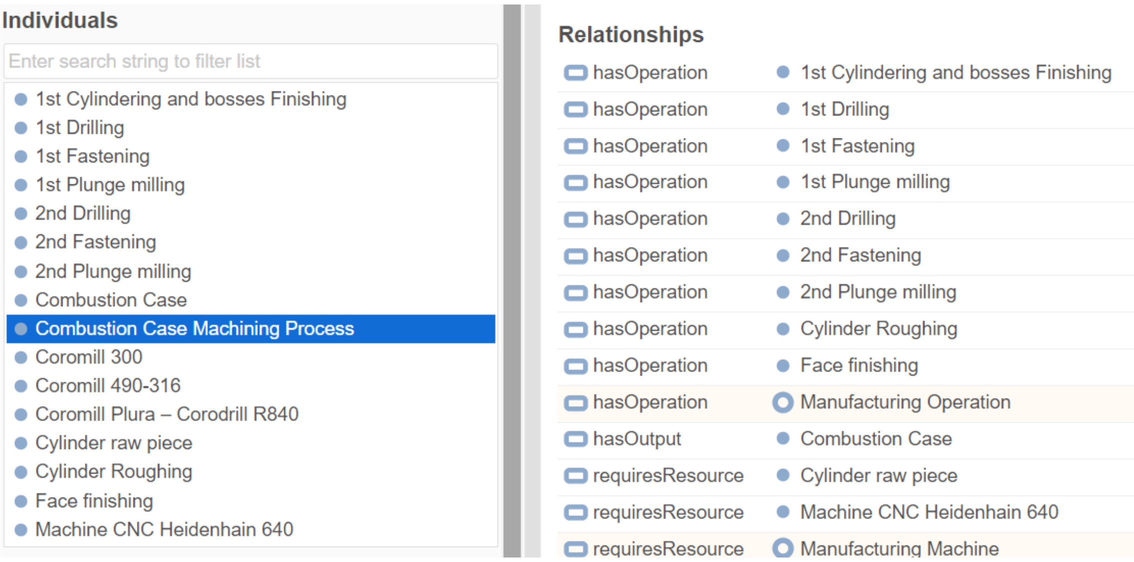

Figure 28 GF pilot ontology individuals and properties.

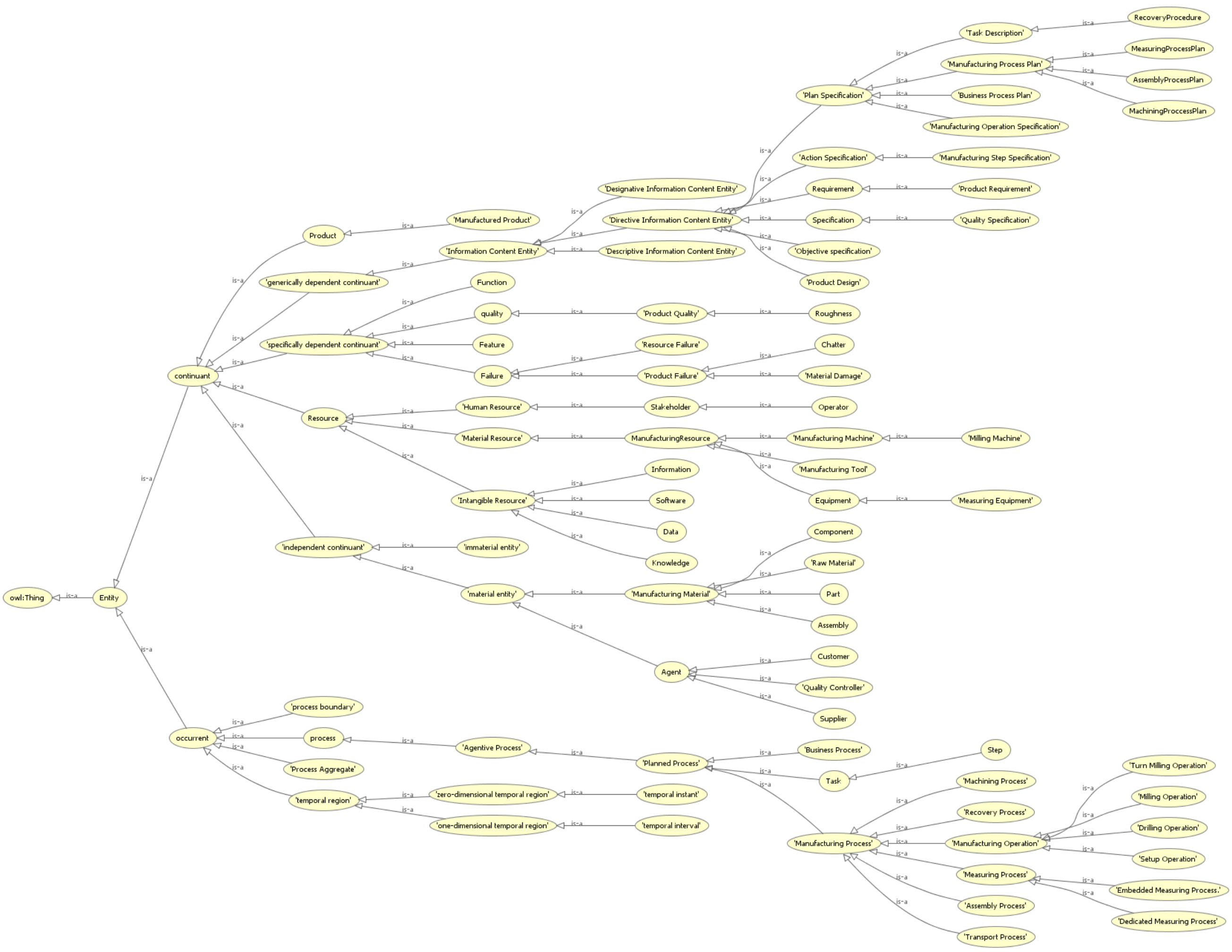

Figure 29 GF pilot application ontology structure and entities (zoom in to check details).

6.4 Access mechanisms for Digital models and Ontologies

To facilitate the reusability, all the models and ontologies described in this report are accessible to relevant stakeholders. The detailed access mechanisms and specifications are introduced as follows. Interested users can follow these approaches to obtain relevant models and ontologies as the basis for new applications.

- ### 6.4.1 Access to the IOF-Core top-level ontology

The IOF-Core ontology is developed by the IOF CORE Working Group[6]. It is still under active development. A repository of the latest version of the Core ontology is shared on Github[7] which is open available.

The repository of IOF-Core ontology contains an OWL format implementation of the top terms, as well as a serialized Turtle format version. Users can directly download this repository and import either the OWL version or Turtle version ontology to an

[6] https://www.industrialontologies.org/top-down-wg/
[7] https://github.com/NCOR-US/IOF-BFO/tree/IOF-Core-2020

ontology development software like Protégé or Webprotégé. It is worth to notice that the IOF-Core ontology is based on BFO. Therefore, the BFO ontology (*bfo-2020.owl*), which is stored in the "imports" folder, has to be imported together with the IOF-Core ontology. Moreover, a SPARQL querying is also included in the repository to support extracting the classes and their annotations.

- ### 6.4.2 Access to the QU4LITY domain ontology

As introduced in previous sections, the QU4LITY Domain ontology is developed following the IOF structure. It is developed using Webprotégé considering its advantages of supporting cooperation among multiple users. This ontology is also under development and might be updated to include feedbacks from application cases. The current version of this ontology is available, read-only, on Webprotégé[8]. For editing and redesign, it can be exported in different format such as RDF, XML, Turtle, OWL etc., as shown in Figure 30. The current version of this ontology in OWL format is attached in a separate document as appendix for readers to get an overview of the ontology.

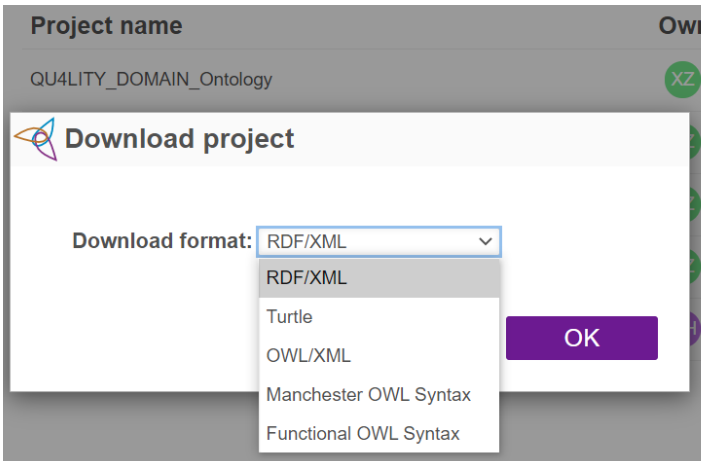

Figure 30 Supported formats for exporting QU4LITY domain ontology

Based on the downloaded QU4LITY domain ontology, more detailed application ontologies can be developed. It is worth to mention a couple of technical details when importing an ontology to Webprotégé. First, the *owl* file which contains the ontology has to be renamed as *"root-ontology.owl"* due to the limitations of Webprotégé. It is not a problem for Protégé desktop version. Second, it is necessary to set the display name properties after imported the ontology. To do this, users need to go to "Project" and "Settings…" menu and then find the "Display Name Settings" section. Make sure the display name settings are set to show *rdfs:label* with *en lang* tag.

- ### 6.4.3 Access to the application ontologies

The application ontologies are developed as demonstrators to showcase how the domain ontology can be applied in industry scenarios. The background and general structures of both application ontologies have been introduced above. Currently they

[8] https://webprotege.stanford.edu/#projects/8b81796d-6c0c-4cfb-875a-df68a444c1af/edit/Classes

are still under frequent update. The current versions are also available on Webprotégé for read-only purpose[9][10]. Due to the large size of these application ontologies, they are not attached as appendix in this report.

[9] https://webprotege.stanford.edu/#projects/cb52a4bf-acff-43e8-957e-9596a069a42d/edit/Classes
[10] https://webprotege.stanford.edu/#projects/6fdf2f03-9f3e-413a-b06d-a9a81f1ec5f0/edit/Classes

7. Conclusions

This deliverable is the final report of the QU4LITY project Work Package (WP) 2 Task T2.5. The topic is "Specification and Prototyping of Digital Models, Vocabularies and Digital Twins".

In this report, we firstly generally introduced the background and objectives of QU4LITY projects and the role of this task under the QU4LITY AQ vision as well as in the reference architecture. Then we reviewed the state-of-the-art of the key concepts, standards, existing projects etc. related to the specification and prototyping of semantic models and ontology engineering for the QU4LITY projects. Based on the review results, the fundamental principles and the methodology for conduction this task are defined. Following these principles and methodology, we analyzed the user stories and stakeholders' requirements, relevant existing top-level and domain ontologies, as well as some relevant standards. Based on the analysis results, we defined the QU4LITY vocabularies.

Moreover, we reviewed existing data models which can be reused for developing the QU4LITY digital models. In particular, the MPFQ model and Z-BREAK model from previous projects are reviewed. Based on these models, we designed the data models for AQ scenario including the RMPFQ model and the semantic-driven digital twin model. To realize the semantic-driven approach, we developed multiple ontologies covering top-level, domain-level and application-level ontologies.

This report mainly focuses on the theoretical part of the project. Some activities about the applications of the digital models and ontologies are still under implementation in several pilots. The final output of the implementation will be presented in the final report of the pilot package.

List of figures

List of abbreviations

AAS: Asset Administration Shell

AQ: Autonomous Quality

AutomationML: Automation Markup Language

B2MML: Business To Manufacturing Markup Language

BFO: Basic Formal Ontology

BLD: Basic Logic Dialect

BOM: Bill of materials

CCO: Common Core Ontologies

CAEX: Computer Aided Engineering Exchange

COLLADA: COLLAborative Design Activity

CRUD: Create, Read, Update and Delete

CT: Cognitive Twins

DAML: DARPA Agent Markup Language

DARPA: U.S. Defense Advanced Research Project Agency

DLO: Domain Level Ontologies

DR: Design Rationale

ERP: Enterprise Resource Planning

GRACE: EU-project "inteGration of pRocess and quAlity Control using multiagent technologies"

HTML: Hypertext Markup Language

IDS: International Data Spaces

IOF: Industrial Ontologies Foundry

KBE: Knowledge-based engineering

KG: Knowledge Graph

KR: Knowledge Representation

MAS: Multi-Agent System

MES: Manufacturing Execution Systems

MESA: Manufacturing Enterprise Solutions Association

MLO: Middle Level Ontologies

MIMOSA: Machinery Information Management Open Systems Alliance

MPFQ: Material, Production Process, Product Functions/Features, Product Quality

OCES: Ontology Commons EcoSystem

OIL: Ontology Interchange Language

OKBC: Open Knowledge Base Connectivity

OML: Ontology Markup Language

OSLC: Open Source Lifecycle Collaboration

OWL: Web Ontology Language

PATO: Phenotypic Trait Ontology

PRD: Production Rule Dialect

PSS: Product Service System

Q-RA: Qu4lity Reference Architecture

RAMI 4.0: Reference Architecture Model for Industrie 4.0

RDF: Resource Description Framework

RIF: Rule Interchange Format

RuleML: Rule Markup Language

SCM: Supply Chain Management

SFC: Sequential Function Chart

SHOE: Simple HTML Ontology Extension

SWRL: Semantic Web Rule Language

TLO: Top Level Ontologies

TRO: Top Refence Ontology

USM: User Story Mapping

WP: Work Package

XSD: XML Schema language

XOL: XML-based Ontology Exchange Language

ZDM: Zero Defect Manufacturing

References

1. Psarommatis F, May G, Dreyfus PA, Kiritsis D. Zero defect manufacturing: state-of-the-art review, shortcomings and future directions in research. International Journal of Production Research. 2020 Jan 2;58(1):1-7.

2. Schumacher H, Johnsson C. ,Communication through B2MML–is that possible. The World Batch Forum North American Conference Chicago, IL May; 2004.

3. Schleipen M, Drath R, Sauer O. ,The system-independent data exchange format CAEX for supporting an automatic configuration of a production monitoring and control system. 2008 IEEE International Symposium on Industrial Electronics: IEEE; 2008.

4. Schleipen M, Okon M. ,The CAEX tool suite-User assistance for the use of standardized plant engineering data exchange. 2010 IEEE 15th Conference on Emerging Technologies & Factory Automation (ETFA 2010): IEEE; 2010.

5. Schleipen M, Sauer O. ,Flexibility and interoperability face Manufacturing Execution Systems (MES). Proceedings of 3rd International Swedish Production Symposium (SPS), Day 2 Book; 2009.

6. PLCopen. Higher efficiency in your application software development.https://www.plcopen.org/ .

7. Estévez E, Marcos M, Lüder A, et al. ,PLCopen for achieving interoperability between development phases. 2010 IEEE 15th Conference on Emerging Technologies & Factory Automation (ETFA 2010): IEEE; 2010.

8. PLCopen. What is PLCopen.https://dwww.plcopen.org/node/52/ .

9. International Organization for Standardization. 2012.ISO/PAS 17506:2012 Industrial automation systems and integration -- COLLADA digital asset schema specification for 3D visualization of industrial data.https://www.iso.org/standard/59902.html .

10. Khronos Group. COLLADA Overview.https://www.khronos.org/collada/ .

11. Diankov R, Ueda R, Okada K, et al. ,Collada: An open standard for robot file formats. Proceedings of the 29th Annual Conference of the Robotics Society of Japan, AC2Q1–5; 2011.

12. Drath R, Luder A, Peschke J, et al. ,AutomationML-the glue for seamless automation engineering. 2008 IEEE International Conference on Emerging Technologies and Factory Automation: IEEE; 2008.

13. AutomationML consortium. 2018.Whitepaper AutomationML Edition 2.1 2018.https://www.automationml.org/o.red/uploads/dateien/1542621846-Whitepaper%20AutomationML%20Edition%202.1%202018.zip .

14. MTConnect. A free, open standard for the factory.https://www.mtconnect.org/ .

15. Vijayaraghavan A, Sobel W, Fox A, et al.,. Improving machine tool interoperability using standardized interface protocols: MT connect 2008.

16. Open Standards for Physical Asset Management. 2019.What is MIMOSA?https://www.mimosa.org/ .

17. El Kadiri S, Kiritsis D. Ontologies in the context of product lifecycle management: state of the art literature review. International Journal of Production Research. 2015 Sep 17;53(18):5657-68.

18. Gómez-Pérez A, Suárez-Figueroa MC,. NeOn Methodology for Building Ontology Networks: a Scenario-based Methodology 2009.

19. Gruber TR,. A translation approach to portable ontology specifications. Knowledge acquisition 1993; 5(2):199-220.

20. Calero C, Ruiz F, Piattini M. ,Ontologies for software engineering and software technology: Springer Science & Business Media; 2006.

21. Mizoguchi R, Ikeda M,. Towards ontology engineering. Journal-Japanese Society for Artificial Intelligence 1998; 13:9-10.

22. Noy NF, Sintek M, Decker S, et al.,. Creating semantic web contents with protege-2000. IEEE intelligent systems 2001; 16(2):60-71.

23. Boley H, Paschke A, Shafiq O. ,RuleML 1.0: the overarching specification of web rules. International Workshop on Rules and Rule Markup Languages for the Semantic Web: Springer; 2010.

24. Lloyd JW. ,Foundations of logic programming: Springer Science & Business Media; 2012.

25. Ron Rudnicki (CUBRC I),. An Overview of the Common Core Ontologies 2019 12 February.

26. Arp R, Smith B,. Function, role and disposition in basic formal ontology 2008.

27. Smith B, Grenon P,. The Cornucopia of Formal-Ontological Relations. Dialectica 2004; 58(3):279-296.

28. ETSI I,. ETSI GS CIM 004 (v1.1.1) 2018.

29. Milicic A, Perdikakis A, El Kadiri S, et al. ,Towards the definition of domain concepts and knowledge through the application of the user story mapping method. IFIP International Conference on Product Lifecycle Management: Springer; 2012.

30. Grüninger M, Fox MS,. Methodology for the design and evaluation of ontologies 1995.

31. Noy NF, McGuinness DL,. Ontology development 101: A guide to creating your first ontology 2001.

32. Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A.,. Ontology library - Wonderweb deliverable D18. https://www.dropbox.com/s/nnynu3r9a02r846/Masolo_et_al-2003-.pdf?dl=0.

33. Guizzardi G, Wagner G, Almeida JPA, et al.,. Towards ontological foundations for conceptual modeling: The unified foundational ontology (UFO) story. Applied ontology 2015; 10(3-4):259-271.

34. Nicola Guarino and Emilio M. Sanfilipo,. Characterizing IOF top terms with the DOLCE (and UFO) ontologies - V1.1 2019 January 31.

35. Epple U, Mertens M, Palm F, et al.,. Using properties as a semantic base for interoperability. IEEE Transactions on Industrial Informatics 2017; 13(6):3411-3419.

36. Foehr M, Jäger T, Turrin C, et al.,. Implementation of a methodology for consideration of product quality within discrete manufacturing. IFAC Proceedings Volumes 2013; 46(9):863-868.

37. O'sullivan A, Sheffrin SM. ,Prentice Hall Economics: Principles in Action: Pearson/Prentice Hall; 2007.

38. Hammer E,. House of Quality (HoQ). Karl-Franzens-Universität Graz.Available online at http://www-classic.unigraz.at/inmwww/NEU/lehre/pdf/Hammer_House_of_Qu ality.pdf, checked on 2011; 21(03):2011.

39. Lu, Jinzhi, Xiaochen Zheng, Ali Gharaei, Kostas Kalaboukas, and Dimitris Kiritsis. "Cognitive twins for supporting decision-makings of Internet of Things systems." In Proceedings of 5th International Conference on the Industry 4.0 Model for Advanced Manufacturing, pp. 105-115. Springer, Cham, 2020.

Partners

Appendix A. State-of-the-art

A.1 Existing industrial data standards and architectures

The flow and exchange of digital data across different ZDM equipment and processes is crucial to the implementation of AQ concept in practice. To realize such data flow and exchange, certain data models and common vocabularies are essential in QU4LITY project. In previous projects and studies, a series of standards for representing plants, production processes and quality processes information have been created. These existing standards should be the basis of specifications for the future data models and vocabularies of QU4LITY project. Some important standards, including B2MML, CAEX, PLCOpen, COLLADA, AutomationML, MTConnect and MIMOSA, are reviewed in this section. Some existing reference architecture like the International Data Spaces (IDS) is also introduced in this section.

• A.1.1 Industrial data exchange standards

▪ *B2MML*

B2MML stands for Business To Manufacturing Markup Language. It is an XML implementation of the ANSI/ISA-95 family of standards, known internationally as IEC/ISO 62264. B2MML consists of a set of XML schemas written using the World Wide Web Consortium's XML Schema language (XSD) that implement the data models in the ISA-95 standard. B2MML is published by the Manufacturing Enterprise Solutions Association (MESA) and it is free to use provided credit is given to MESA.

B2MML can help companies to integrate business systems, such as Enterprise Resource Planning (ERP) and Supply Chain Management (SCM) systems, with manufacturing systems, such as Manufacturing Execution Systems (MES) following ISA-95. Figure 31 [2] shows a part of the B2MML schema for Production Performance. Although B2MML is useful to exchange information between the business system and the manufacturing and control system, there also exist several limitations. For example, it requires both systems are compliant with the ISA S95 standard and support XML [2].

▪ *CAEX*

Computer Aided Engineering Exchange (CAEX) is an abstract object-oriented data format based on XML, which depicts real or logical plant objects in form of data objects. It allows the predefinition of pattern solutions in form of classes which can be instantiated several times [3]. CAEX contains an XML meta model for describing the setup and structure of plant data. The format supports library concepts and object-oriented approaches. It is possible to integrate libraries from users and suppliers as well as project libraries. In addition, both a top-down and a bottom-up system design are supported. It enables the syntactic and semantic unification of the data which allows the required configuration algorithms to be decoupled from the data sources [3,4]. CAEX consists of three types of libraries [4]: InterfaceClassLibraries, RoleClassLibraries and SystemUnitClassLibraries. In addition to these libraries, there is the InstanceHierarchy where the specific plant is modelled. These main components of CAEX form the basis for a CAEX file. In addition, there are further elements which serve the purpose of detail specification or definition of links between the elements. Figure 32 shows a basic model of CAEX [5].

```
<xsd:complexType name = "ProductionPerformance">
  <xsd:sequence>
      <xsd:element name = "ID"                      type = "IDType"/>
      <xsd:element name = "Description"             type = "DescriptionType"
                                   minOccurs = "0" maxOccurs = "unbounded"/>
      <xsd:element name = "Location"                type = "LocationType"
                                                    minOccurs = "0"/>
      <xsd:element name = "PublishedDate"           type = " PublishedDate Type"
                                                    minOccurs = "0"/>
      <xsd:element name = "ProductionScheduleID"    type = " ProductionScheduleIDType"
                                                    minOccurs = "0"/>
      <xsd:element name = "startTime"               type = " startTimeType"
                                                    minOccurs = "0"/>
      <xsd:element name = "EndTime"                 type = "EndTimeType"
                                                    minOccurs = "0"/>
      <xsd:element name = "EquipmentElementLevel"   type = " EquipmentElementLevelType"
                                                    minOccurs = "0"/>
      <xsd:element name = "ProductionResponse"      type = "ProductionResponseType"
                                                    minOccurs = "0" maxOccurs = "unbounded"/>
      <xsd:ant namespace="##any"
                                                    minOccurs = "0" maxOccurs = "unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

Figure 31 B2MML schema for Production Performance

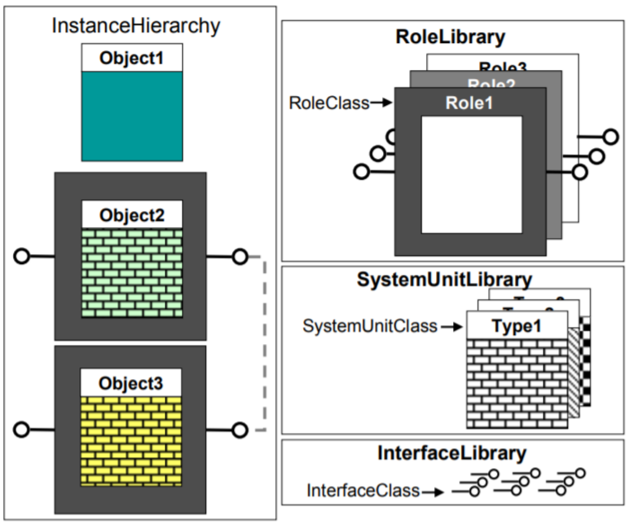

Figure 32 Structure of CAEX model

▪ ***PLCOpen***

PLCopen is a vendor and product independent worldwide association. The aim of this association consists on resolving topics related to control programming to support the use of international standards in this field [6]. It promotes the use of IEC 61131–3 standard for industrial control programming. The use of this standard in industrial control field provides a standard programming interface, so it allows people with different background and expertise to create different parts of a program automation project during different phases of the development life cycle. The standard includes the definition of the Sequential Function Chart (SFC) language, used to structure the internal organization of a program, and four programming languages; two textual: Instruction List, and Structured Text and other two graphics: Ladder Diagram, Function Block Diagram [7].

Current topics of PLCOpen include: Motion Control and Safety functionality, XML data exchange format standardizing the base data of IEC projects in software systems, and mapping to the OPC Unified Architecture for transparent Communication. In the future, the practical automation tasks will be challenged by new industry demands and new products and PLCopen will remain focusing on global harmonization [8].

▪ ***COLLADA***

COLLAborative Design Activity (COLLADA) is an interchange file format for interactive 3D applications. It defines an XML-based schema to enable 3D authoring applications to freely exchange digital assets without loss of information, enabling multiple software packages to be combined into extremely powerful tool chains [9]. The intermediate language provides comprehensive encoding of visual scenes including geometry, shaders and effects, physics, animation, kinematics, and even multiple version representations of the same asset. COLLADA FX enables leading 3D authoring tools to work effectively together to create shader and effects applications and assets to be authored and packaged using OpenGL® Shading Language, Cg, CgFX, and DirectX® FX [10].

The main features of COLLADA include [11]:

- core content management tags dictating the referencing structure,
- graphics scene extensions allowing complex shaders and geometries,
- simple rigid-body dynamics parameters,
- boundary representation models,
- and kinematics structures.

▪ ***AutomationML***

Automation Markup Language (AutomationML) is a data exchange format based on XML schema and it was developed to support the data exchange in a heterogeneous engineering tools landscape. It provides concepts to store engineering information about topology, geometry, kinematics, logic, references and relations following the object-oriented paradigm. It allows modelling of physical and logical plant components as data objects encapsulating different aspects. An object may consist of other sub-objects and may itself be part of a larger composition or aggregation [12,13].

AutomationML combines existing industry data formats that are designed for the storage and exchange of different aspects of engineering information. These data formats are used on an "as-is" basis within their own specifications. The core of AutomationML is the top-level data format CAEX which is further introduced in the following section. CAEX is utilized to interconnect the different data formats, which enables a distributed document architecture. Figure 33 [13] illustrates the basic architecture of AutomationML. It consists of the standards CAEX, PLCopen XML and COLLADA. CAEX acts as the top-level format and stores the plant topology, COLLADA stores geometric and kinematic information, while PLCopen XML serves for the storage of sequences and behavior.

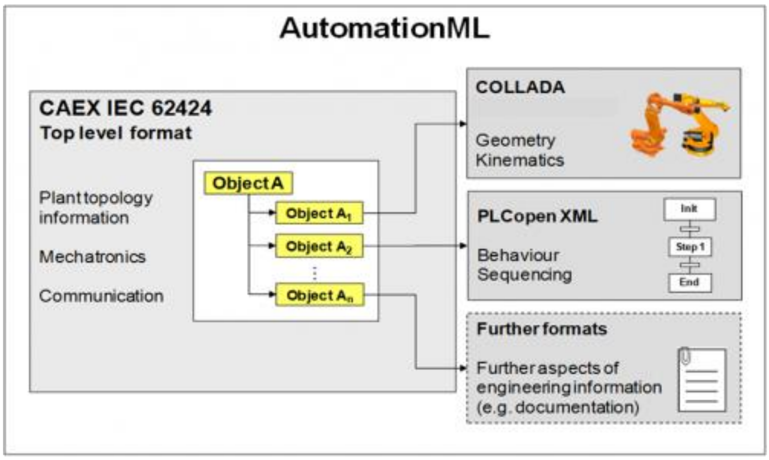

Figure 33 Basic architecture of AutomationML

The main advantages of AutomationML include [12]:

- Reuse of matured data formats - this reduces the specification effort for AutomationML.
- Distribution of data into different files – this eases the handling of bulk data.
- Simplified usage of library files – they can be stored and exchanged separately.
- Different geometry or logic variants may be stored separately, e.g. in order to distinguish between different degrees of detail.

▪ ***MTConnect***

The MTConnect standard offers a semantic vocabulary for manufacturing equipment to provide structured, contextualized data with no proprietary format. With uniform data, developers and integrators can focus on useful, productive manufacturing applications rather than translation. MTConnect data sources include things like production equipment, sensor packages, and other hardware. Applications using MTConnect data provide more efficient operations, improved production optimization, and increased productivity. MTConnect provides domain-specific vocabulary and data models, is extensible, and integrates with other standards by design [14]. MTConnect is not intended to replace the functionality of existing products, but it strives to enhance the data acquisition capabilities of devices and applications and move toward a plug-and-play environment to reduce the cost of integration. It provides a high level of interoperability with other standards and tools. It supports vast amount of data and information types including [15]:

- Physical and actual device design data
- Measurement or calibration data
- Near-real-time data from the device

A schematic of a factory system with MTConnect integration is shown in Figure 34 [15].

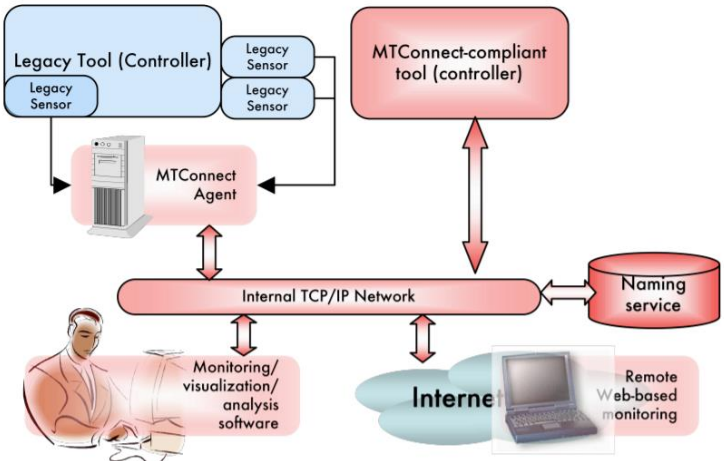

Figure 34 A schematic of a factory system with MTConnect integration

- ***MIMOSA***

Machinery Information Management Open Systems Alliance (MIMOSA) is a not-for-profit industry trade association dedicated to developing and encouraging the adoption of open, supplier-neutral IT and IM standards enabling physical asset lifecycle management spanning manufacturing, fleet and facilities environments. MIMOSA standards support key requirements for Critical Infrastructure Management on a cross-sector basis, addressing the highly heterogeneous and interdependent nature of critical infrastructure. MIMOSA standards and collaboratively developed specifications enable a Digital Twin to be defined and maintained on a supplier-neutral basis, while also using that Digital Twin to provide Context for Big Data (IIOT and other sensor-related data) and Analytics [16].

A.1.2 The International Data Spaces (IDS)

The IDS[11] is a virtual data space using standards and common governance models to facilitate the secure exchange and easy linkage of data in business ecosystems. It thereby provides a basis for creating and using smart services and innovative

[11] https://internationaldataspaces.org/

business processes, while at the same time ensuring digital sovereignty of data owners.

The reference architecture model of IDS consists of the following four architectures:

- The business architecture addresses questions regarding the economic value of data, the quality of data, applicable rights and duties (data governance), and data management processes,
- The security architecture addresses questions concerning secure execution of application software, secure transfer of data, and prevention of data misuse,
- The data and service architecture specify (in an application and technology independent form) the functionality of the Industrial Data Space, especially the functionality of the data services, on the basis of existing standards (vocabularies, semantic standards etc.),
- The software architecture specifies the software components required for pilot testing of the Industrial Data Space. Existing technologies are being used as far as possible.

The reference architecture model thereby serves as a blueprint for different implementations of the Industrial Data Space. Both the research project and the user association are eager to get in touch with similar projects and initiatives.

One application example of IDS is the end-to-end monitoring of goods during transportation in the supply chain management scenario. In many industries, such as the pharmaceutical and the chemical industry, the products need to be transported under guarantee of special precautions only, as otherwise they would be damaged or destroyed.

Unfavorite ambient conditions, such as temperature humidity, shock, vibration etc., may pose a multitude of risks to sensitive goods. These ambient conditions can be monitored during transportation by means of sensors, and the respective data can be transmitted via mobile radio communication. Thereby potential risks can be detected early enough, and appropriate measures for risk reduction can be taken more quickly. The IDS serves as a platform for customers and suppliers allowing end-to-end monitoring of ambient conditions goods are exposed to during transportation. Customers and suppliers are provided with data necessary to be informed at any time as to where certain goods are at a certain moment and in what condition these goods are. In doing so, the Industrial Data Space ensures that companies receive all data required, while at the same time ensuring data sovereignty on the part of the company sending the data.

• A.1.3 The GAIA-X project

GAIA-X[12] is a project initiated by Europe for Europe and beyond. Its aim is to develop common requirements for a European data infrastructure. Therefore openness, transparency and the ability to connect to other European countries are central to GAIA-X. An open digital ecosystem is needed to enable European companies and

[12] https://www.data-infrastructure.eu/GAIAX/Navigation/EN/Home/home.html

business models to compete globally. This ecosystem should allow both the digital sovereignty of cloud services users and the scalability of European cloud providers. GAIA-X aims to develop the foundations for a federated, open data infrastructure based on European values. GAIA-X connects centralized and decentralized infrastructures in order to turn them into a homogeneous, user-friendly system. The resulting federated form of data infrastructure strengthens the ability to both access and share data securely and confidently.

GAIA-X is set to be an Infrastructure and Data Ecosystem according to European values and standards. This overall mission drives its architecture.1 The architecture employs digital processes and information technology to facilitate the interconnection between all participants in the European digital economy. By leveraging existing standards, open technology and concepts, it enables open, consistent, quality-assured and easy-to-use innovative data exchange and services. Additionally, GAIA-X will become a facilitator for interoperability and interconnection between its Participants, for data as well as services.

GAIA-X builds on a unique selection of technological approaches to bring digital sovereignty to life, including federation, self-descriptions and policies, identity and trust. The GAIA-X architecture principles are as follows:

- Openness and Transparency: The specification and documentation of GAIA-X technologies and architectures will be accessible to GAIA-X Participants worldwide. The technical steering and roadmap of GAIA-X is done in public and the involvement of private sector players is disclosed. Technology choices will be made in order to encourage distribution of collaboratively created artifacts under open source licenses,
- Interoperability: All GAIA-X Participants will be able to interact with each other in a well-specified way. This architecture describes the technical means to achieve that but is agnostic to and works beyond specific implementations,
- Federated Systems: GAIA-X specifies federated systems of autonomous Providers, tied together by a specified set of standards, frameworks, and legal rules. The federation supports decentralization and distribution,
- Authenticity and Trust: An identity management system with mutual authentication, selective disclosure, and revocation of trust is needed to foster a secure digital ecosystem without building upon the authority of a single corporation or government.

In order to fulfill its vision and principles, the GAIA-X architecture imposes technical guidelines:

- Security-by-design: GAIA-X puts security technology at its core to protect every Participant and system who is part of a GAIA-X eco system,
- Privacy-by-design: this architecture fundamentally considers all privacy-related aspects,
- Enabling federation, distribution and decentralization: it is not a goal to build up centralized, homogeneous, isolated solutions. Instead, this architecture considers approaches like federation, distribution and decentralization,

- Usage-friendliness and simplicity: State-of-the-art user experience, open standards and protocols, and streamlined processes will be crucial for GAIA-X adoption and acceptance,
- Machine-Processability: All GAIA-X artifacts are machine readable. For the exchange of these artifacts, systems expose as the primary means of interaction in GAIA-X. Human User Interfaces will leverage APIs to enable the interaction of humans with GAIA-X. Automation is supported by this architecture,
- Semantic representation: By building on machine processability, it is ensured that a GAIA-X data model is established, which carries the semantics of the ecosystem and effectively delivers interoperability. Core elements for semantic representation are policy requirements and Self-Descriptions, enabling the translation of actual use cases into digital processes.

The high-level overview of the GAIA-X architecture is shown in Figure 35[13].

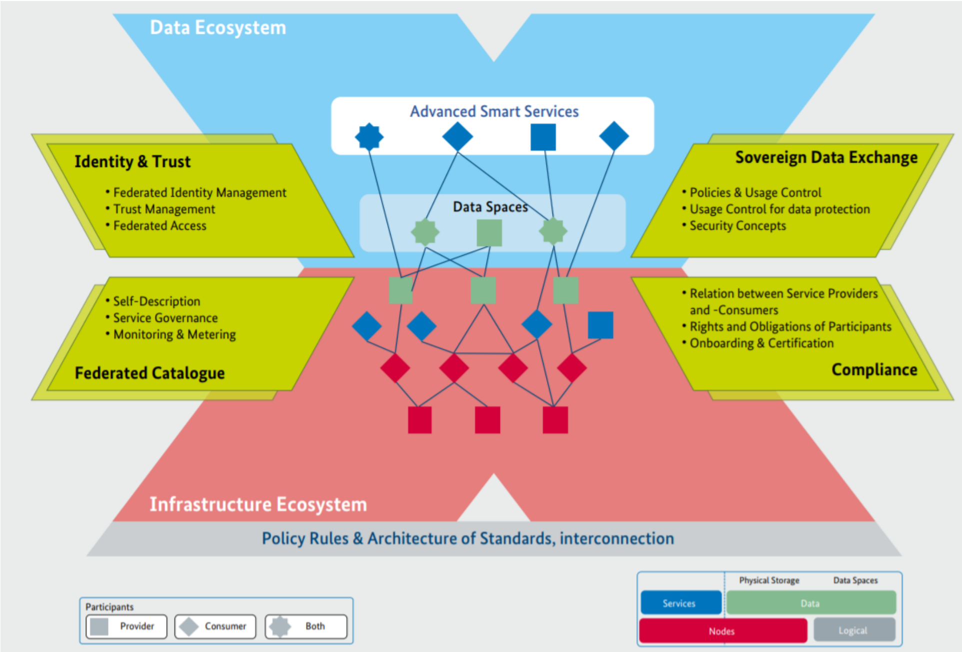

Figure 35 High-level overview of the GAIA-X architecture.

A.2 Ontology engineering and semantic modelling

Data from different platforms and sources might be heterogeneous in syntax, schema, or semantics, which make data integration and data interoperability difficult. Ontology engineering and semantic modelling provide solutions to achieve semantic interoperability in a heterogeneous information system. The following sections introduce the concepts of ontology engineering and semantic modelling, as well as

[13] https://www.data-infrastructure.eu/GAIAX/Redaktion/EN/Publications/gaia-x-technical-architecture.pdf?__blob=publicationFile&v=5

some relevant standards and languages. Several existing ontologies are reviewed in the end of this chapter.

- **A.2.1 Ontology Engineering**

Ontology has been a popular concept in philosophy, computer and information science for several decades. It is considered as a promising methodology for the future progress in artificial intelligence as it depends on massive formalization of the ontological features of the common-sense physical reality. Ontology in this context is focused on capturing information about the world that is compatible with the perspective of human common sense. In the field of information systems, ontology is mainly used as a reference schema providing a unified and coherent view over existing systems. More recently, ontology has gained importance and popularity with the advent of the Semantic Web. In this context, as to obtain an automated access to the information contained in the Web, information items are described by means of metadata provided by an ontology [17].

Ontology development has become an engineering discipline, Ontology Engineering, which refers to the set of activities that concern the ontology development process and the ontology lifecycle, the methods and methodologies for building ontologies, and the tool suites and languages that support them Ontology engineering is the general term of methodologies and methods for building ontologies. Ontology engineering refers to "The set of activities that concern the ontology development and the ontology lifecycle, the methods and methodologies for building ontologies and the tool suites and languages that support them." [18] The results of ontology engineering provide domain knowledge representation to be reused efficiently and prevent waste of time and money which are usually caused by non-shared knowledge. It helps Information Technology (IT) to operate with interoperability and standardization.

Ontology can play one or several of these roles simultaneously for data, information and knowledge management [17]:

1) Trusted source of knowledge: Ontology provides the representation of concepts, their properties and relationships, in addition to the axioms and rules. Ontology that does not represent all the semantics of a domain through a considerable number of axioms and rules considered lightweight ontology or rather taxonomy. Whether it is lightweight or heavyweight ontology, we consider that the main and basic role of ontology is the provision of a trusted and common source of knowledge used and shared by human or/and software agents,
2) Database: Ontology in the first role as a trusted source of knowledge is considered as a reference framework. The resulting logic-based representations form a conceptual model that can help with storage, management and sharing of data. Thus, ontology can play an additional role that is the role of a database. The conceptual model is preserved and stored along with real data: the instances. Concepts, relationships, properties and instances are considered as unique resources having all a unique resource

identifier. These characteristics particularity confer to the ontology the characteristic of being machine-understandable unlike the logic representation of traditional relational databases that does not specify the same level of semantics. This is due to the mechanisms applied while converting the conceptual model to physical and logical models. This conversion results in semantic loss in information representation, which requires a preliminary knowledge of the conceptual model in order to operate on the database,

3) Knowledge base: Ontology rules and axioms defining the semantics of a domain are represented with logic-based languages. They are considered as intentional or explicit knowledge. Being machine-understandable, the ontology can simultaneously play an additional role of a knowledge base and support the deduction of implicit knowledge by processing rules using an inference engine. Rule inference engines can for instance chain several rules and generate more complex conclusions,
4) Bridge for multiple domains: Eliminating the need for repetition of design process for every application domain is worthy of consideration and can be possible by leveraging external resources. Through manual, semi-automatic or automatic ontology-based mechanisms such as mapping, alignment, specialization and merging, it is possible to adopt and extend existing ontological resources and metadata initiatives therefore bridging multiple domains from design, manufacturing, assembly, etc. A number of technics and approaches to ontology merging and alignment exist in the literature. Besides, having domain information and knowledge structured into ontology leads to avoid a number of transformations of knowledge from one formalism to another,
5) Mediator for interoperability: As a reference framework, ontology can serve as a basis for schema matching to support systems interoperability in close environments where systems, tools and data sources have no common recognition of data type and relationships,
6) Contextual search enabler: Ontology can play the role of an enabler of a contextual search engine answering complex and cross-domain questions without any specific knowledge on the data source using a common language for querying. Ontology will enable retrieving contextual relationships behind an entity, avoiding thus unnecessary and irrelevant data. Contextual information can be also extracted through the application of rules to derive the relevancy of context elements in specific situations,
7) Linked Data enabler: The application of ontologies has grown with the advent of the Linked Data paradigm. Linked Data consist of the creation of data stores, called triple stores, using URIs for identifying resources and their relations. The application of the Linked Data principles appears to be a very promising approach in data integration.

- **A.2.2 Semantic Modelling**

Ontology represents the nature of being, becoming, existence, and so on in the way of philosophy. One of the most well-known is: "ontology is an explicit, formal specification of a shared conceptualization of a domain of interest" [19].

Semantic modelling can help defining the data and the relationships between entities. [20] An information model provides the ability to abstract different kind of data and provides an understanding of how the data elements are related. A semantic model is a type of information model that supports the modelling of entities and their relationships. The total set of entities in a semantic model comprises the taxonomy of classes that can be used to represent the real world.

The main objective of semantic modelling techniques is to define the meaning of data within the context of its correlation, and to model the domain world in the abstract level. The benefits of exploiting semantic data models for business applications are mainly as follows:

- Avoiding misunderstanding: by providing a clear, accessible, agreed set of terms, relations as a trusted source and discussions, misunderstandings can easily be resolved.
- Conduct reasoning: by being machine understandable and through the usage of logic statements (rules), ontologies enable automatic reasoning and inference which leads to automatic generation of new and implicit knowledge.
- Leverage resources: by extending and relating an application ontology to external ontological resources, via manual or automatic mapping and merging processes, the need for repetition of entire design process for every application domain is eliminated.
- Improve interoperability: semantic models can serve as a basis for schema matching to support systems' interoperability in close environments where systems, tools and data sources have no common recognition of data type and relationships.

Ontologies provide formal models of domain knowledge exploited in different ways. Therefore, ontology plays a significant role for many knowledge-intensive applications. Depending on corresponding languages, several different knowledge representation formalisms exists. However, they share a common set of components such as classes, relations, formal axioms and instances.

- Classes represent concepts, which are taken in a broad sense. For instance, in the Product Lifecycle domain, concepts are: Life Cycle phase, Product, Activity, Resources, Event, and so on. Classes in ontology are usually organized in taxonomies through which inheritance mechanisms can be applied.
- Relations represent a type of association between concepts of the domain. They are formally defined as any subset of a product of *n* sets, that is: $R \subset C_1 x C_2 x ... x C_n$. Ontologies usually contain binary relations. The first argument is known as the domain of the relation, and the second argument is the range.
- Formal axioms serve to model sentences that are always true. They are normally used to represent knowledge that cannot be formally defined by the other components. In addition, formal axioms are used to verify the consistency of the ontology itself or the consistency of the knowledge stored in a knowledge base. Formal axioms are very useful to infer new knowledge.

- Instances are used to represent elements or individuals in an ontology.

As a Design Rationale (DR), ontology can be used as follows [21]:

- Level 1: Used as a common vocabulary for communication among distributed agents.
- Level 2: Used as a conceptual schema of a relational database. Structural information of concepts and relations among them is used. Conceptualization in a database is nothing other than conceptual schema. Data retrieval from a database is easily done when there is an agreement on its conceptual schema.
- Level 3: Used as the backbone information for a user of a certain knowledge base. Levels higher than this plays role of the ontology, which has something to do with "content".
- Level 4: Used for answering competence questions.
- Level 5: Standardization
  - Standardization of terminology (at the same level of Level 1)
  - Standardization of meaning of concepts
  - Standardization of components of target objects (domain ontology).
  - Standardization of components of tasks (task ontology)
- Level 6: Used for transformation of databases considering the differences of the meaning of conceptual schema. This requires not only the structural transformation but also semantic transformation.
- Level 7: Used for reusing knowledge of a knowledge base using DR information.
- Level 8: Used for reorganizing a knowledge base based on DR information.

- ### A.2.3 Standards and Languages for Semantic Web

  - #### *Ontology languages*

Ontology Markup Language (OML)

OML was developed at the University of Washington, is partially based on SHOE. In fact, it was first considered an XML serialization of SHOE. Hence, OML and SHOE share many features. Four different levels of OML exist: OML Core is related to logical aspects of the language and is included by the rest of the layers; Simple OML maps directly to RDF(S); Abbreviated OML includes conceptual graphs features; and Standard OML is the most expressive version of OML. We selected Simple OML, because the higher layers don't provide more components than the ones identified in our framework. These higher layers are tightly related to the representation of conceptual graphs. There are no other tools for authoring OML ontologies other than existing general-purpose XML edition tools.

XML-based Ontology Exchange Language (XOL)

The US bioinformatics community designed XOL for the exchange of ontology definitions among a heterogeneous set of software systems in their domain. Researchers developed it after studying the representational needs of experts in bioinformatics. They selected Ontolingua (a Tool for Collaborative Ontology Construction) and OML as the basis for creating XOL, merging the high expressiveness of OKBC-Lite, a subset of the Open Knowledge Based Connectivity protocol, and the syntax of OML, based on XML. There are no tools that allow the development of ontologies using XOL. However, since XOL files use XML syntax, we can use an XML editor to author XOL files.

Simple HTML Ontology Extension (SHOE)

SHOE is a small extension to HTML which allows web page authors to annotate their web documents with machine-readable knowledge. SHOE makes real intelligent agent software on the web possible. HTML was never meant for computer consumption; its function is for displaying data for humans to read. The "knowledge" on a web page is in a human-readable language (usually English), laid out with tables and graphics and frames in ways that we as humans comprehend visually. Unfortunately, intelligent agents aren't human. Even with state-of-the-art natural language technology, getting a computer to read and understand web documents is very difficult. This makes it very difficult to create an intelligent agent that can wander the web on its own, reading and comprehending web pages as it goes. SHOE eliminates this problem by making it possible for web pages to include knowledge that intelligent agents can actually read.

Ontology Interchange Language (OIL)

OIL was developed in the OntoKnowledge project (www.ontoknowledge.org/OIL), permits semantic interoperability between Web resources. Its syntax and semantics are based on existing proposals (OKBC, XOL, and RDF(S)), providing modeling primitives commonly used in frame-based approaches to ontological engineering (concepts, taxonomies of concepts, relations, and so on), and formal semantics and reasoning support found in description logic approaches (a subset of first order logic that maintains a high expressive power, together with decidability and an efficient inference mechanism). OIL, built on top of RDF(S), has the following layers: Core OIL groups the OIL primitives that have a direct mapping to RDF(S) primitives; Standard OIL is the complete OIL model, using more primitives than the ones defined in RDF(S); Instance OIL adds instances of concepts and roles to the previous model; and Heavy OIL is the layer for future extensions of OIL. OILEd, Protégé2000, and WebODE can be used to author OIL ontologies. OIL's syntax is not only expressed in XML but can also be presented in ASCII. We use ASCII for our examples.

DARPA Agent Markup Language + OIL (DAML+OIL)

DAML+OIL has been developed by a joint committee from the US and the European Union (IST) in the context of DAML, a DARPA project for allowing semantic interoperability in XML. Hence, DAML+OIL shares the same objective as OIL. DAML+OIL is built on RDF(S). Its name implicitly suggests that there is a tight relationship with OIL. It replaces the initial specification, which was called DAML-ONT, and was also based on the OIL language. OILEd, OntoEdit, Protégé2000, and WebODE are tools that can author DAML+OIL ontologies.

Web Ontology Language (OWL)

OWL is the result of the work of the W3C Web Ontology Working Group. This language derived from DAML+OIL and, as the previous languages, is intended for publishing and sharing ontologies in the Web. OWL is built upon RDF(S), has a layered structure and is divided into three sublanguages: OWL Lite, OWL DL and OWL Full. OWL is grounded on Description Logics and its semantics are described in two different ways: as an extension of the RDF(S) model theory and as a direct model-theoretic semantics of OWL. Both have the same semantic consequences on OWL ontologies.

- OWL 2: OWL 2 is an extension and revision of OWL that adds new functionality with respect to OWL; some of the new features are syntactic sugar (e.g., disjoint union of classes) while others offer new expressivity. OWL 2 includes three different profiles (i.e., sublanguages) that offer important advantages in particular application scenarios, each trading off different aspects of OWL's expressive power in return for different computational and/or implementation benefits. These profiles are:
- OWL 2 EL: It is particularly suitable for applications where very large ontologies are needed, and where expressive power can be traded for performance guarantees.
- OWL 2 QL: It is particularly suitable for applications where relatively lightweight ontologies are used to organize large numbers of individuals and where it is useful or necessary to access the data directly via relational queries (e.g., SQL).
- OWL 2 RL: It is particularly suitable for applications where relatively lightweight ontologies are used to organize large numbers of individuals and where it is useful or necessary to operate directly on data in the form of RDF triples. OWL 2 ontologies: The Direct Semantics that assigns meaning directly to ontology structures and the RDF-Based Semantics that assigns meaning directly to RDF graphs.

Resource Description Framework (RDF)

RDF, developed by the W3C for describing Web resources, allows the specification of the semantics of data based on XML in a standardized, interoperable manner. It also provides mechanisms to explicitly represent services, processes, and business models, while allowing recognition of nonexplicit information. The RDF data model is equivalent to the semantic networks formalism. It consists of three object types:

- Resources are described by RDF expressions and are always named by URIs plus optional anchor IDs
- Properties define specific aspects, characteristics, attributes, or relations used to describe a resource
- Statements assign a value for a property in a specific resource (this value might be another RDF statement)

The RDF data model does not provide mechanisms for defining the relationships between properties (attributes) and resources—this is the role of RDFS. RDFS offers primitives for defining knowledge models that are closer to frame-based approaches.

RDF(S) is widely used as a representation format in many tools and projects, such as Amaya, Protégé, Mozilla, SilRI, and so on.

According to W3C, RDF model has advantages as follows:

- The RDF model is made up of triples: as such, it can be efficiently implemented and stored; other models requiring variable-length fields would require a more cumbersome implementation.
- The RDF model is essentially the canonicalization of a (directed) graph and has all the advantages (and generality) of structuring information using graphs.
- The basic RDF model can be processed even in absence of detailed information (an "RDF schema") on the semantics: it already allows basic inferences to take place, since it can be logically seen as a fact basis.
- The RDF model has the important property of being modular.

The union of knowledge (directed graphs) is mapped into the union of the corresponding RDF structures. Since RDF is a standard model for data interchange and is a W3C recommendation designed to standardize the definition and use of metadata-descriptions of Web-based resources, it is well suited to representing data. As knowledge representation, when it comes to semantic interoperability, RDF has significant advantages [22]: The object-attribute structure provides natural semantic units because all objects are independent entities. A domain model—defining objects and relationships—can be represented naturally in RDF. To find mappings between two RDF descriptions, techniques from research in knowledge representation are directly applicable.

- ***Semantic Rule Languages***

Rules are widely recognized to be a major part of the frontier of the Semantic Web, and critical to the early adoption and applications of knowledge-based techniques in e-business, especially enterprise integration and B2B e-commerce. This includes Knowledge Representation (KR) theory and algorithms; mark-up languages based on such KR; engines, translators, and other tools; relationships to standardization efforts; and, not least, applications. Interest and activity in the area of Rules for the Semantic Web has grown rapidly over the last years.

Known rule systems fall into three broad categories: first-order, logic-programming, and action rules. These paradigms share little in the way of syntax and semantics. Moreover, there are large differences between systems even within the same paradigm.

Rule Interchange Format (RIF)

RIF is a W3C supported standard for exchanging rules among rule systems, in particular among Web Rule Engines. RIF is focused on exchange rather than trying to develop a single one-fits-all rule language because, in contrast to other Semantic Web standards, such as RDF and OWL, it is clear by the involved working groups that a single language would not satisfy the needs of many popular paradigms for using rules in knowledge representation and business modeling. But even rule exchange alone is recognized as a daunting task.

Regarding RIF, the approach taken by the Working Group was to design a family of languages, called dialects, with rigorously specified syntax and semantics. The family of RIF dialects is intended to be uniform and extensible. RIF uniformity means that dialects are expected to share as much as possible of the existing syntactic and semantic apparatus. Extensibility here means that it should be possible for motivated experts to define a new RIF dialect as a syntactic extension to an existing RIF dialect, with new elements corresponding to desired additional functionality. These new RIF dialects would be non-standard when defined but might eventually become standards. Because of the emphasis on rigor, the word format in the name of RIF is somewhat of an understatement. RIF in fact provides more than just a format. However, the concept of format is essential to the way RIF is intended to be used. Ultimately, the medium of exchange between different rule systems is XML, a format for data exchange. Central to the idea behind rule exchange through RIF is that different systems will provide syntactic mappings from their native languages to RIF dialects and back. These mappings are required to be semantics-preserving, and thus rule sets can be communicated from one system to another provided that the systems can talk through a suitable dialect, which they both support. The RIF Working Group has focused on two kinds of dialects: logic-based dialects and dialects for rules with actions. Generally, logic-based dialects include languages that employ different types of logic, such as first-order logic (often restricted to Horn logic) or non-first-order logics underlying the various logic programming languages (e.g., logic programming under the well-founded or stable semantics). The rules-with-actions dialects include production rule systems, such as Jess, Drools and JRules, as well as reactive (or event-condition action) rules, such as Reaction RuleML. Due to the limited resources of the RIF Working Group, it defined only two logic dialects, the Basic Logic Dialect (RIF-BLD) and a subset, the RIF Core Dialect, shared with RIF-PRD; the Production Rule Dialect (RIFPRD) is the only rules-with-actions dialect defined by the group. Other dialects are expected to be defined by the various user communities.

Rule Markup Language (RuleML)

RuleML constitutes a family of Web rule languages which contains derivation (deduction) rule languages, which themselves have a webized Datalog language as their inner core. Datalog RuleML's atomic formulas can be (un)keyed and (un)ordered. Inheriting the Datalog features, Hornlog RuleML adds functional expressions as terms. In Hornlog with equality, such misinterpreted (constructor-like) functions are complemented by interpreted (equationdefined) functions. These are described by further orthogonal dimensions "single- vs. setvalued" and "first- vs. higher-order". Combined modal logics apply special relations as operators to atoms with a misinterpreted relation, complementing the usual interpreted ones [23].

RuleML is a markup language developed to express both forward (bottom-up) and backward (top-down) rules in XML for deduction, rewriting, and further inferential-transformational tasks. A number of markup languages that are defined as part of RuleML are the following:

- Mathematical Markup Language17 (MathML)
- DARPA Agent Markup Language18 (DAML)
- Predictive Model Markup Language19 (PMML)
- Attribute Grammars in XML20 (AG-markup)
- Extensible Stylesheet Language Transformations21 (XSLT)

Semantic Web Rule Language (SWRL)

The Semantic Web Rule Language (SWRL) is based on a combination of the OWL DL and OWL Lite sublanguages of the OWL Web Ontology Language with the Unary/Binary Datalog RuleML sublanguages of the Rule Markup Language (RuleML). The proposal extends the set of OWL axioms to include Horn-like rules. It thus enables Horn-like rules to be combined with an OWL knowledge base. A high-level abstract syntax is provided that extends the OWL abstract syntax described in the OWL Syntaxes22 document. An extension of the OWL model-theoretic semantics is also given to provide a formal meaning for OWL ontologies including rules written in this abstract syntax.

The proposed rules are of the form of an implication between an antecedent (body) and consequent (head). The intended meaning can be read as: whenever the conditions specified in the antecedent hold, then the conditions specified in the consequent must also hold. Both the antecedent (body) and consequent (head) consist of zero or more atoms. An empty antecedent is treated as trivially true (i.e. satisfied by every interpretation), so the consequent must also be satisfied by every interpretation; an empty consequent is treated as trivially false (i.e., not satisfied by any interpretation), so the antecedent must also not be satisfied by any interpretation. Multiple atoms are treated as a conjunction. Note that rules with conjunctive consequents could easily be transformed (via the Lloyd-Topor transformations) [24]into multiple rules each with an atomic consequent.

Atoms in these rules can be of the form C(x), P(x,y), sameAs(x,y) or differentFrom(x,y), where C is an OWL description, P is an OWL property, and x,y are either variables, OWL individuals or OWL data values. A XML syntax is also given for these rules based on RuleML and the OWL XML presentation syntax. Furthermore, an RDF concrete syntax based on the OWL RDF/XML exchange syntax is presented. The rule syntaxes are illustrated with several running examples.

- ### A.2.4 Top-level Ontologies and relevant domain ontologies

  - ***Top-level ontologies***

Top-level ontologies are ontologies which consist of very general terms (such as "object", "property", "relation") that are common across all domains. An important function of an top-level ontology is to support broad semantic interoperability among a large number of domain-specific ontologies by providing a common starting point for the formulation of definitions. Terms in the domain ontology are ranked under the terms in the upper ontology, e.g., the top-level ontology classes are superclasses or supersets of all the classes in the domain ontologies. For QU4LITY ontologies, three top-level ontologies are considered including BFO, CCO and IOF. IOF is the latest one which takes BFO and CCO as foundations.

Basic Formal Ontology (BFO)

The BFO is a small, upper-level ontology that is designed for use in supporting information retrieval, analysis and integration in scientific and other domains. The aim of BFO is to facilitate interoperability among domain ontologies that are built in its terms through a process of downward population. The structure of BFO is based on a division of entities into two disjoint categories of continuant and occurrent, the former comprehending for example objects and spatial regions, the latter comprehending processes conceived as extended through (or as spanning) time. BFO thereby seeks to incorporate both three-dimensionalist and four-dimensionalist perspectives on reality within a single framework. The general overview of the class hierarchy of BFO including all classes is shown in Figure 36. There are more explanations about how BFO will used as the basis for the development of QU4LITY ontologies.

- ENTITY
  - CONTINUANT
    - INDEPENDENT CONTINUANT
      - IMMATERIAL ENTITY
        - SITE
        - SPATIAL REGION
          - ZERO-DIMENSIONAL SPATIAL REGION
          - ONE-DIMENSIONAL SPATIAL REGION
          - TWO-DIMENSIONAL SPATIAL REGION
          - THREE-DIMENSIONAL SPATIAL REGION
        - FIAT CONTINUANT BOUNDARY
          - ZERO-DIMENSIONAL CONTINUANT FIAT BOUNDARY
          - ONE-DIMENSIONAL CONTINUANT FIAT BOUNDARY
          - TWO-DIMENSIONAL CONTINUANT FIAT BOUNDARY
      - MATERIAL ENTITY
        - OBJECT
        - OBJECT AGGREGATE
        - FIAT OBJECT
    - SPECIFICALLY DEPENDENT CONTINUANT
      - QUALITY
      - REALIZABLE ENTITY
        - DISPOSITION
          - FUNCTION
        - ROLE
    - GENERICALLY DEPENDENT CONTINUANT
  - OCCURRENT
    - PROCESS
    - PROCESS BOUNDARY
    - TEMPORAL REGION
      - ZERO-DIMENSIONAL TEMPORAL REGION
      - ONE-DIMENSIONAL TEMPORAL REGION
    - SPATIOTEMPORAL REGION

Figure 36 General overview of the class hierarchy of BFO.

Common Core Ontologies (CCO)

The CCO comprise eleven ontologies that aim to represent and integrate taxonomies of generic classes and relations across all domains of interest. Accompanying these ontologies is a rule-based method for representing the content of any data source whatsoever through constructing domain ontologies as extensions of CCO [25].

CCO is designed as a mid-level extension of BFO and the Relation Ontology (RO), an upper-level ontology framework widely used to structure and integrate ontologies in the biomedical domain [26]. BFO aims to represent the most generic categories of entity, and RO the most generic types of relations that hold between them, by defining a small number of classes and relations. CCO then extends from BFO-RO in the sense that every class in CCO is asserted to be a subclass of some class in BFO, and that CCO adopts the generic relations defined in RO (*e.g., has_part*) [27]. Accordingly, CCO classes and relations are heavily constrained by the BFO-RO framework, from which it inherits much of its basic semantic relationships. The eleven mid-level ontologies that comprise the CCO include:

- Information Entity Ontology
- Agent Ontology
- Quality Ontology
- Event Ontology
- Artifact Ontology
- Time Ontology
- Geospatial Ontology
- Units of Measure Ontology
- Currency Unit Ontology
- Extended Relation Ontology
- Modal Relation Ontology

The content of each of these ontologies is built within the upper-level semantic framework defined by BFO-RO. This means that the basic class hierarchy, as well as many relationships, are defined by BFO and merely inherited by CCO. Consequently, compliance with the semantics of the CCO requires compliance with the semantic framework defined by BFO-RO.

Among the eleven mid-level ontologies, the Quality Ontology represents the attributes of agents, artifacts, and events. These attributes change over time and are often used to differentiate objects from others of the same or similar type. Since attributes are always dependent on other entities (their bearers), the classes contained in the Quality Ontology are only of value when used in combination with classes from other ontologies. Much of the content is adapted from the Phenotypic Trait Ontology (PATO) and extends the BFO classes: QUALITY, REALIZABLE ENTITY (parent class of DISPOSITION and ROLE), and PROCESS PROFILE. Subclasses of qualities in Quality Ontology include: SHAPE QUALITY, WEIGHT, and TEMPERATURE. Subclasses of REALIZABLE ENTITY include: MAGNETISM, COLOR, and VULNERABILITY. Subclasses of PROCESS PROFILE include: SPEED and FREQUENCY. Thus, the Quality Ontology enables one to represent the qualities of a person or the speed of a patrol ship, as shown in Figure 37 and Figure 38.

CCO provide the means to express complex relationships that other OWL-based vocabularies cannot. They were developed under the adherence of principles designed to maximize their ability to provide interoperability and reduce the costs associated with organizing enterprise information. They are grounded in doctrine, vetted against data, and subjected to quality tests. But most importantly the ontologies provide a starting point on which enterprise data interoperability can be built [25].

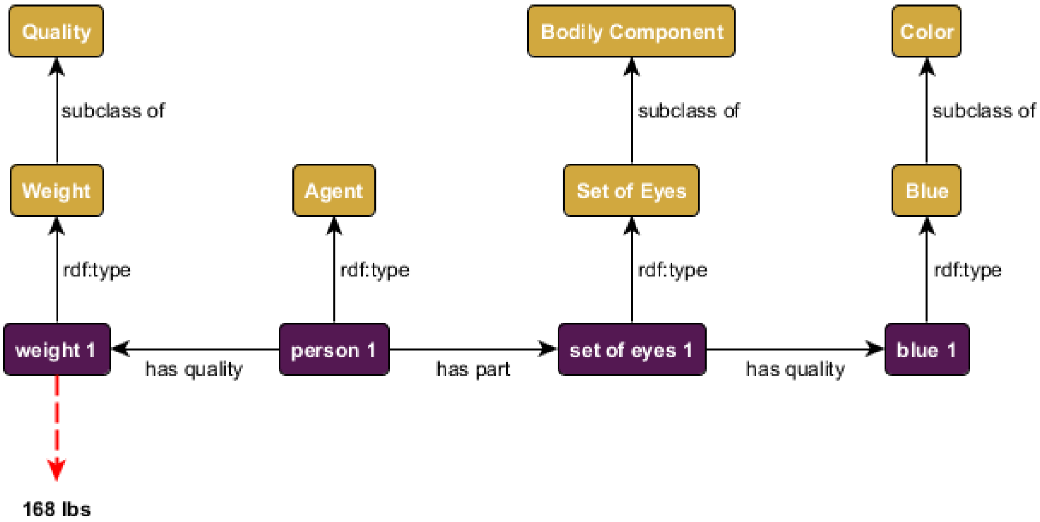

Figure 37 Attributes of a person based on the CCO Quality Ontology.

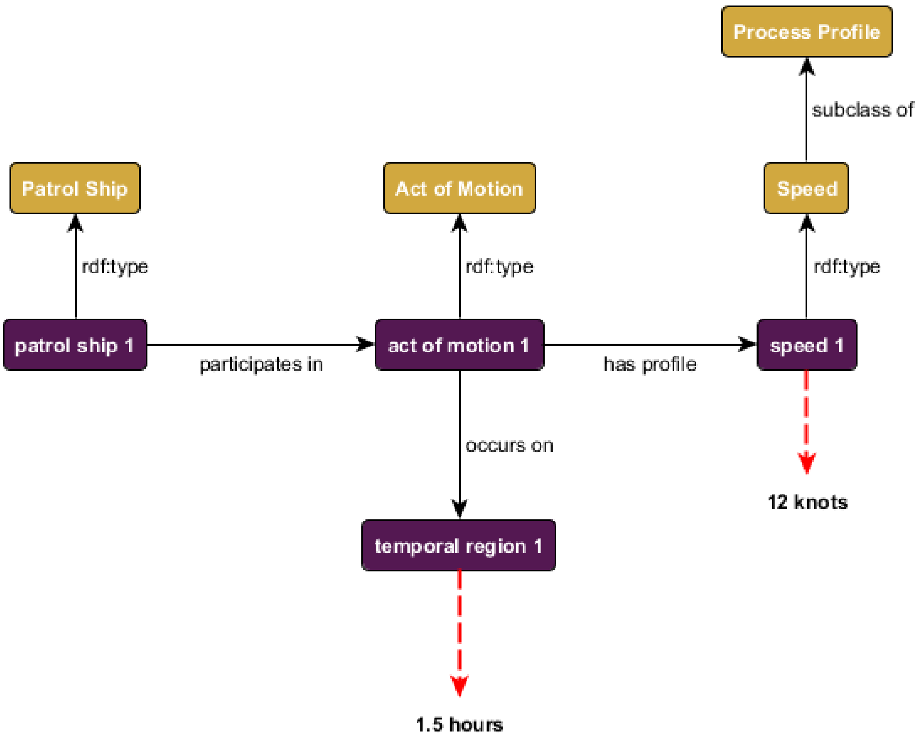

Figure 38 Attributes of a moving ship based on the CCO Quality Ontology.

Industrial Ontologies Foundry (IOF)

In some previous and on-going projects, some domain ontologies related to advanced manufacturing have been developed. They can be used as the references and basis for the development of QU4LITY ontologies.

The IOF is a group working to co-create a set of open ontologies to support the manufacturing and engineering industry needs and advance data interoperability. It involves government, industry, academic and standards organizations to advance data interoperability in their respective fields. The main goals are:

- Creating a suite of open and principles-based ontologies, from which other domain dependent or application ontologies can be derived in a modular fashion, remaining 'generic' (i.e., non-proprietary, non-implementation specific) so they can be reused in any number of industrial domains or manufacturing specializations.
- Providing principles and best practices by which quality ontologies can be developed that will support interoperability for industrial domains,
- Instituting a governance mechanism to maintain and promulgate the goals and principles,
- Providing an organizational framework and governance processes that ensure conformance to principles and best practices for development, sharing, maintenance, evolution, and documentation of IOF ontologies.

For QU4LITY ontologies, IOF provides a general architecture to guide the ontology development process. Figure 39 shows the IOF Ontology-Network architecture.

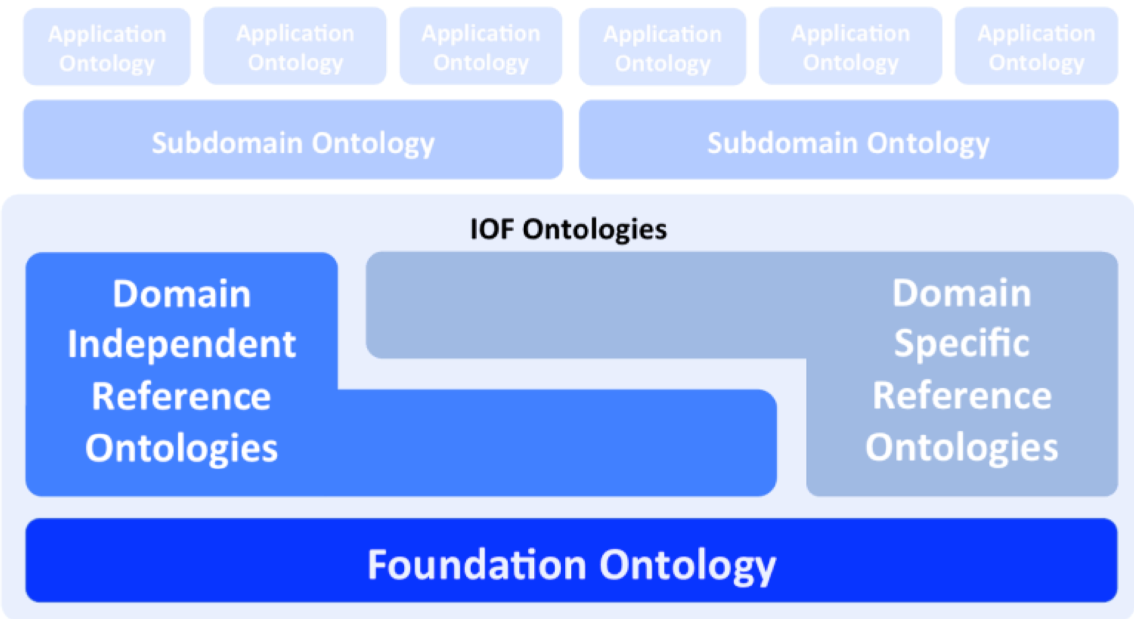

Figure 39 IOF architecture vision[14].

[14] https://www.industrialontologies.org/technical-principles/

- ***Domain ontologies***

IOF-PSS domain ontology

The IOF-PSS working group is developing a domain ontology for Product Service Systems (PSS) design and implementation. Modern industrial companies aim to extend their products with services as fundamental value-added activities and reduce the product to be just a part of the offering. One of the key potentials of such services, besides radical improvements in the use of products, is a reduction of environmental footprint of products and services. To support dynamic building and deployment of new services around products, i.e. to build PSS, there is a need for strong collaboration among various actors across the value chain. Building services today is connected with adding/upgrading of cyber-physical features, as for example adding sensors and intelligence to the products which can be used for various services. This in turn requires dynamic feedback loops between the design, manufacturing and product-service use. Real time exchange of knowledge between the designers, manufacturers, maintenance experts. This includes automatic data gathering and exchange along the value chain (e.g. data on energy consumption for manufacturing of product and/or for its use), but also tacit knowledge from various actors (e.g. experience of the maintenance staff or shop-floor workers on the manufacturability of a part). The ontologies are key tools for effective knowledge sharing in development, deployment and use of PSS in the manufacturing industry.

The IOF-PSS ontology is a joint approach of the two EU projects, FALCON and DIVERSITY, both dealing with PSS in the manufacturing industry. They both investigate methods and tools for PSS development, deployment and use:

- FALCON (Feedback mechanisms Across the Lifecycle for Customer-driven Optimization of iNnovative product-service design) aims to deploy user experiences and data collected via the Internet of Things (IoT) and social media, for the improvement of product-service systems (PSS). Customers play no longer a passive role in the product and service development process, as they express their product and service experiences and opinions through social media. In addition, sensor systems in combination with products incorporated in the IoT, are becoming increasingly common. The potential endless amounts of available information offer a rich ground for value creation in the product-service innovation chain. Accordingly, FALCON develops a Virtual Open Platform to seamlessly connect product-service usage information to design and development processes.
- DIVERSITY (Cloud Manufacturing and Social Software Based Context Sensitive Product-Service Engineering Environment for Globally Distributed Enterprise) aims at developing a new cloud-based engineering environment to support modern enterprises in managing their multi-directional exchange of knowledge, and their dynamic and real-time feedback loops, both internally (among product design, service design and manufacturing) and externally (with customers, both business customers and individual customers - consumers, suppliers and other relevant organizations, across the value chain, distributed all over the globe). In this sense, DIVERSITY aims at providing a

> concurrent collaborative environment for product-service design, supporting companies from the context-sensitive capturing and searching of knowledge, to the transformation of these data into product-service functionalities. In particular, the project ambition is to support those companies that are changing their business model, by extending their products-offerings with services and that are adopting the new mass-customization paradigm.

Following the BFO as upper ontology, the classes of IOF-PSS ontology are defined to clearly distinguish between the so-called Continuant and Occurrent entities. For example, the class Service is defined, under Occurrent, as a process which is executed in time (i.e. a specific service with duration), while the class Service Offer under Continuant is used, without focusing on the temporal aspect, to describe the service as a whole, or as a model, which needs to be planned / designed / developed / etc., and might be affected by other entities like the products, actors, etc. The relations between such pairs of Occurrent and Continuant, like for example *Service* implements *Service Offer*, or *Service is_instance_of/occurent_part_of Service Offer*, are also defined.

The proposed PSS ontology shares a number of classes with other ontologies relevant for manufacturing, such as Product, Service, Company, Resources, etc. Therefore, the objective was not to further elaborate such generic classes in the manufacturing domain, but to indicate those that of relevance for PSS in the manufacturing industry. These classes serve as ‘interfaces’ to other ontologies and will need to be harmonised with these ontologies under the scope of IOF. Therefore, for the current draft version, such classes, like Product and Product Component, and their relations, are not further elaborated or defined in detail (e.g. for Product Component the Product Component Role is not defined etc.)

Smart Appliances REFerence (SAREF)

The SAREF ontology is a shared model of consensus that facilitates the matching of existing assets (standards/protocols/datamodels/etc.) in the smart appliance domain. The SAREF ontology provides building blocks that allow separation and recombination of different parts of the ontology depending on specific needs. The starting point of SAREF is the concept of device (e.g., a switch). Devices are tangible objects designed to accomplish a particular task in households, common public buildings or offices. In order to accomplish this task, the device performs one or more functions. For example, a washing machine is designed to wash (task) and to accomplish this task it performs the start and stop function. The SAREF ontology offers a list of basic functions that can be eventually combined in order to have more complex functions in a single device. For example, a switch offers an actuating function of type 'switching on/off'. Each function has some associated commands, which can also be picked up as building blocks from a list. For example, the 'switching on/off' is associated with the commands 'switch on', 'switch off' and 'toggle'. Depending on the function(s) it accomplishes, a device can be found in some corresponding states that are also listed as building blocks. When connected to a network, a device offers a service, which is a representation of a function to a network that makes the function discoverable, registerable and remotely controllable by other devices in the network. A service can represent one or more functions. A service is offered by a device that wants (a certain set of) its function(s) to be discoverable, registerable, remotely controllable by other devices in the network. A service must specify the device that is offering the service and the function(s) to be represented. A device in the SAREF ontology is also characterized by a profile that can be used to optimize some property, such as Energy, in a home or office that are part of a building.

NGSI-LD

NGSI-LD is a simple yet powerful standard open API for Context Information enabling close to real-time access to information coming from many different sources (not only IoT data sources) [28]. The aim of NGSI-LD is to make it easier for end-users, city databases, Internet-of-Things and third-party apps to exchange information.

The NGSI-LD Information Model is defined at two levels: the foundation classes which correspond to the Core Meta-model, as shown in Figure 40, and the Cross-Domain Ontology, as shown in Figure 41. The former amounts to a formal specification of the "property graph" model. The latter is a set of generic, transversal classes which are aimed at avoiding conflicting or redundant definitions of the same classes in each of the domain-specific ontologies. Below these two levels, domain-specific ontologies or vocabularies can be devised. For instance, the SAREF Ontology mentioned above can be mapped to the NGSI-LD Information Model, so that smart home applications will benefit from this Context Information Management API specification [28].

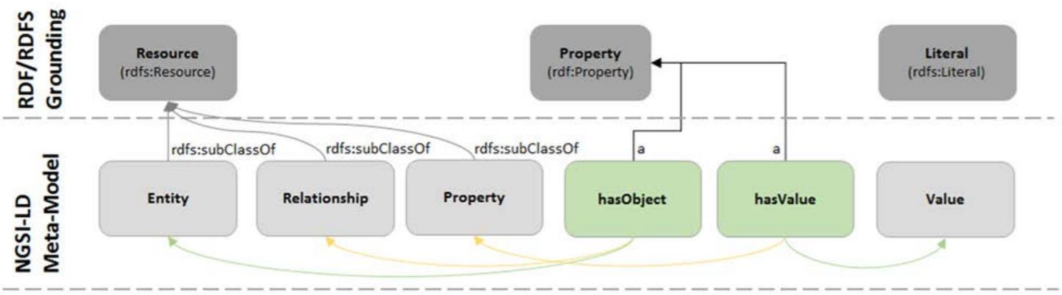

Figure 40 NGSI-LD Core Meta-Model.

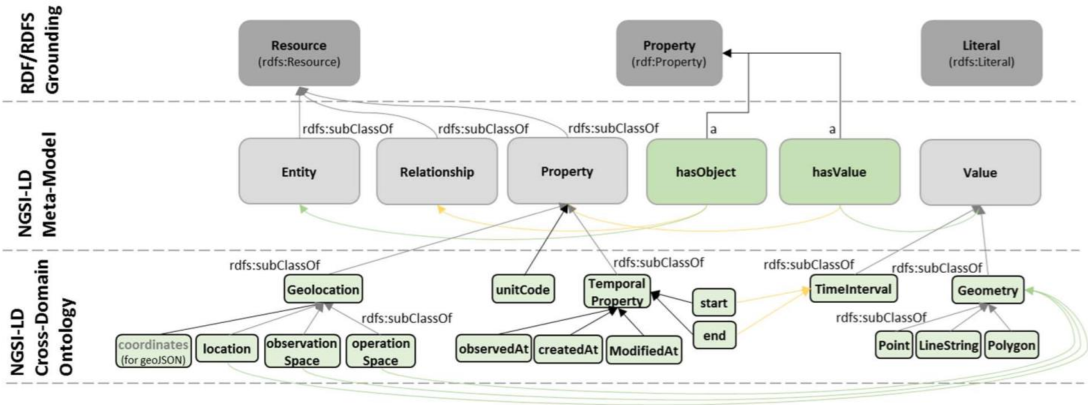

Figure 41 NGSI-LD Core Meta-Model plus the Cross-Domain Ontology.

- ***The OntoCommons project***

The OntoCommons project[15] (Ontology-driven data documentation for Industry Commons) aims to bring together and coordinate activities of the most relevant EU stakeholders for the development of an Ontology Commons EcoSystem (OCES), consisting of ontologies and tools following specific standardization rules, that can be effectively used as foundation for data documentation in the industrial domain, in order to facilitate data sharing and valorization and overcome the existing interoperability bottlenecks.

The targeted OCES covers different levels of ontologies and specification etc. The main components include:

- an OntoCommons Top Refence Ontology (TRO), in the form of a selected set of existing and widely used Top Level Ontologies (TLO) made of a mutual set of alignments between the selected TLOs (i.e. axioms providing correspondences between entities of TLOs), that will formally constitute the apical point of the hierarchy. An ontology alignment approach (the so-called harmonisation) will maximize the use of existing domain ontologies developed under different TLOs.
- Middle Level Ontologies (MLO), to allow smooth connections between TLOs, lower level ontologies and commonly needed entities such as time, information, unit, space etc.
- Domain Level Ontologies (DLO), as needed by demonstrators, both harmonised existing domain ontologies and newly developed domain ontologies, following the develop/test/validate/agree procedure.
- EcoSystem Requirements and Specifications, to ensure homogeneity between ontologies becoming part of the OES, such as formalization in specific ontology language and documentation.
- Tools, a selected set of tools for the practical implementation of data documentation and its exploitation, that are ready to be used with ontologies respecting OCES requirements.

The overview of the OCES is shown in Figure 42. It consists of:

- a hierarchy of networked ontologies of different levels of generality (from top-level to application level) for which multiple forms of interoperability will be provided
- a set of tools and methodologies selected from the available state of the art, covering the full range of OntoCommons activities, from ontology development (e.g. editors) to reasoning (e.g. reasons) and database integration.
- a set of specifications for ontologies that will provide full compatibility between tools and ontologies.

The OCES will adopt a pluralist approach for the ontological representation of a domain of interest, meaning that more than one ontology for the same domain may

[15] https://ontocommons.eu/

be hosted. While a monistic approach would simplify data sharing and harmonization, the pluralist approach is by far more inclusive with respect to users and will enable reuse of already available resources.

Currently, the OntoCommons project is ongoing ad some of the QU4LITY partners are involved. The results obtained in this project will be harmonized with the OntoCommons activities.

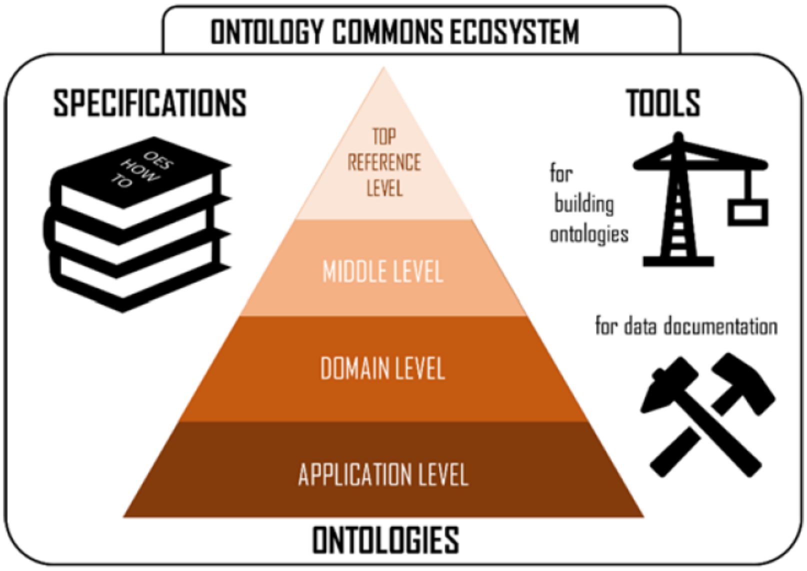

Figure 42 Ontology Commons EcoSystem overview.

Appendix B. QU4LITY top terms and definitions

No.	RMPFQ Terms		References IOF/GRACE
	1st level	2nd level	
1	Material		GRACE:Material - entity used during the production process, e.g. tubs, blocks of steel, bearings, nuts and bolts, according to the BOM.
1.1		Raw Material	IOF:RawMaterial - A portion of matter from which physical artifacts can be made.
1.2		Assembly	IOF: Assembly - A material entity (BFO: object aggregate) that is designed for a particular purpose and is a collection of two or more interconnected components. [IOF informal definition: An assembly is a combination of parts and components that form a functional entity.]
1.3		Component	IOF: Component - A material entity (BFO: object) that is designed for a particular purpose and intended to become a part of another material entity (e.g., rivet is intended to be used in assembly of the airframe) IOF informal definition: A component is a part or subassembly that goes into a higher-level assembly or the final product (adapted from APICS). Explanation: A particular artifact can be considered as a component or an assembly depending on the context of a manufacturing process.
1.4		Part	IOF: Component part (TBD)
2	Process	Planed process	BFO: Process (Elucidation) p is a process =Def p is an occurrent that has some temporal proper part and for some time t, p has some material entity as participant
2.1		Business Process	IOF: Business Process - A structured set of activities performed to achieve an organizational objective.
2.2		Manufacturi ng Process	IOF: Manufacturing Process - A structured set of manufacturing activities performed to achieve the organizational objective of producing or refining a product or service, that generally consumes raw materials and components and uses manpower, tools and other equipment in order to fulfill its purpose. (Note: This term may well be modeled a subclass of OrganizationalProcess.) GRACE: ProcessPlan - Represents the manufacturing process to produce a product
3	Func tion		GRACE: Function - Entity that describes interactions among product components (materials) and/or external environment, e.g. tub contains water and drum move clothes; it is a product function.

			IOF informal definition: A design is a specification that describes a collection of features to be created for a product. IOF: Design - A representation of the collection of features that satisfy requirements.
4	Feature		IOF: Feature IOF:PSS: Feature - Feature is PSS Information that describes the characteristics of a product, service, or PSS. IOF:PSS: ProductFeature - Product Feature is a Feature that describes an aspect, or attribute, related to the product within the PSS scope. GRACE: Property - an attribute that characterizes a resource (i.e. a skill) or that a resource should satisfy to execute an operation (i.e. a requirement). It includes a mathematical operator associated to the property value, e.g. a speed equal to 2000 rpm.
5	Failure		
5.1		Product Failure	
5.2		Resource Failure	GRACE: Failure - description of an occurred perturbation, including the occurrence date, the applied recover procedure and the recovery time.
6	Manufacturing Resource		IOF: Manufacturing Resource - A person or material entity that is available participates in the manufacture of a product or component. (Any entity that can assume a Manufacturing Resource Role.) GRACE: Resource - entity that can execute a certain range of operations as long as its capacity is not exceeded. Producer, quality controller, transporter, operator and tool are specializations of resource and inherit its characteristics.
6.1		Manufacturing Machine	IOF: Manufacturing Machine - A piece of equipment that uses power to apply forces and/or control movement that can bear a Manufacturing Role GRACE: Producer - a specialized resource entity that is responsible for the execution of producing operations, such as a welding robot or a CNC (Computer Numerical Control) machine.
6.2		Manufacturing Tool	IOF: Manufacturing Tool - A piece of equipment used by a machine or as stand-alone and can bear a Manufacturing Role GRACE: Tool - a specialized resource entity representing the physical devices used by producer stations and by operators to execute their processing operations, e.g. screw driver for

			screwing the counterweights; it may include the physical devices used by transporter resources to execute their handling operations, e.g. grippers.
6.3		Equipment	IOF: Equipment - Any Artifact or collection of related Artifacts that can bear a Manufacturing Role. GRACE: Transporter (Mover) can be considered as a subclass of Equipment?
7	Product		IOF: Product - Product is an item or substance that is a result of some processes GRACE: Product - economic entity (finished or semi-finished), which is produced by the enterprise in a value-adding process (it includes a Bill of Materials (BOM), i.e. the list of materials that are considered as components of a final or intermediate product; it also includes the quantity of each material required).
8	Product Quality		IOF: Product Quality - A physical or functional characteristic of a product that can be measured or qualitatively evaluated.
9	Plan		Plan specification =def. directive ICE with some action specification and some objective specification as parts.
9.2		Manufacturing Process Plan	IOF: Manufacturing Process Plan - A specification that prescribes the collection of related activities in a manufacturing process that produces a product. Source: from IOF collective discussions GRACE: Journal - description of the production of a product instance belonging to a production order executed in the production line, including the list of operations performed and the resources that have executed each operation.
9.3		Operation Specification	IOF: Operation Specification -An information artifact that groups an ordered set of Step Definitions to achieve part of a Manufacturing Process and specifies the type of Resources needed to perform the work. GRACE: Operation - a job executed by one resource like drilling, welding, assembly, inspection and maintenance, that may add value to the product or may measure the value of the product, e.g. the quality control.
9.4		Recovery Procedure	GRACE: Recovery Procedure - entity that describes the procedure to recover from the occurrence of a failure.
9.5		Step	IOF: Step - An information artifact that is part of an Operation Definition and prescribes an action to be taken by a worker at a fine-grained level of detail.

9.6		Task	IOF: Task - An information artifact that prescribes some part of a Plan [that can be performed by a resource].