YRKESHÖGSKOLAN NOVIA

A Data Model for Engine Laboratory Measurements

Creation of UML Map for Laboratory Engine Test Measurements Database based on ASAM ODS standard.

Francesco Trevisan

Degree Thesis

Thesis for a Master of Engineering (UAS) - degree

Industrial Management and Engineering

Vaasa 2024

DEGREE THESIS

Author: Francesco Trevisan

Degree Programme and place of study: Industrial Management and Engineering, Vaasa

Specialisation: Master in Industrial Management and Engineering (UAS) - degree

Supervisor(s): Ray Pörn

Title: Engine Laboratory Measurements Data Model

Date: 01.06.2024 Number of pages: 36

Abstract

Marine engine tests in laboratory creates tens of thousands data for each test run, which are then stored and used to analyze and improve engine performances. The implementation of more flexible, lighter and precise database would improve the data analysis and traceability enabling more precise understanding and simulation of engine behavior. As a matter of fact in the last years, automotive industry has been deploying similar databases in its testing procedure, and established a standardized approach for these solution, named ASAM ODS.

The main purpose of this thesis work has been creating the data model, more specifically the UML map, of the future database based on ASAM ODS standards. The work has involved the collection of the information about current data acquisition, storage and control inside Wärtsilä company and the shaping of the data model based on these assumptions. The resulting data model has been described and commented, justifying the choices taken and highlighting the benefits of the solution proposed.

Language: English

Key Words: ASAM ODS, Measurement Database, UML Map.

Table of Contents

1 Introduction

The Engine Laboratory Measurement Database Project has born in 2022 as an internal development project of Wärtsilä Finland, Finnish company where the candidate is currently employed. This project has been built on the perspective of creating a new functional tool for data collection, storing and analysis, the first brick for the construction of a machine learning system for engine development in Wärtsilä. The main objective of the work was to understand the current way of working and generate a tailored data model, specifically a UML map, by which the new engine laboratory database could be built in the next future months and years.

In the following chapters a short introduction of the problem will be given, starting from the company and engine laboratory description and proceeding with a deeper description of the project and the work.

1.1 Company presentation

Wärtsilä is a global leader in smart technologies and complete lifecycle solutions for the marine and energy markets. With a strong focus on sustainability and innovation, Wärtsilä provides advanced solutions that enable its customers to achieve their environmental and operational goals. The company's diverse portfolio includes engines, propulsion systems, energy storage, and digital solutions, all designed to optimize performance, efficiency, and reliability. With a presence in over 200 locations worldwide, Wärtsilä is committed to creating a sustainable future for the marine and energy industries.

Wärtsilä offers a wide range of products for the marine industry. Some of the main products include:

- Engines: Wärtsilä manufactures and supplies both medium-speed and high-speed engines for various marine applications; an example of Wärtsilä engine is shown in Figure 1.
- Propulsion Systems: Wärtsilä provides propulsion solutions such as controllable pitch propellers (CPP), fixed pitch propellers (FPP), and waterjets.
- Electrical & Automation Systems: Wärtsilä offers electrical and automation systems for marine vessels, including power distribution, vessel control, navigation, and communication solutions.

- Environmental Solutions: Wärtsilä is committed to sustainability and offers exhaust gas cleaning systems (scrubbers), ballast water management systems, and waste treatment systems.
- LNG Solutions: Wärtsilä is a leading provider of liquefied natural gas (LNG) solutions for the marine industry; this includes LNG fuel systems, LNG bunkering solutions, and LNG regasification systems.
- Services: Wärtsilä provides a comprehensive range of services to support its marine customers throughout the lifecycle of their equipment. This includes maintenance, repairs, spare parts, technical support, and performance optimization services.

Figure 1. Examples of Wärtsilä engine W20V31SG.

Assembling Wärtsilä engines can be a complex process due to the intricate design and high level of precision required. The engines consist of numerous components that need to be carefully assembled and aligned to ensure optimal performance and reliability. Skilled technicians follow detailed assembly procedures and use specialized tools and equipment to complete the process accurately. The complexity can vary depending on the specific engine model and its configuration. Such complexity influences directly the optimization of engines performances. Each component plays a crucial role in the overall performance of the engine, and any changes or adjustments made to one component can have a ripple effect on others (e.g. vibration level, combustion stability, etc.). This interdependency requires a comprehensive understanding of the engine's design and operation.

Furthermore, engines operate under a wide range of conditions, including varying loads, speeds, and environmental factors, all interconnected between each others. Optimizing performance requires considering these factors and finding the right balance between efficiency, power output, emissions, and durability, all depending on specific technical parameters such as timing, fuel pressure, exhaust gas temperatures and many others. Additionally, optimizing engine performance often involves trade-offs between different parameters. For example, improving fuel efficiency may require sacrificing some power output, or reducing emissions may impact other performance aspects. Balancing these trade-offs requires careful consideration and expertise. Furthermore, engine optimization is an ongoing process that evolves with advancements in technology, regulations, and customer requirements. Continuous research and development are necessary to stay at the forefront of engine performance optimization.

Such level of complexity, hardware and performances, require extensive engine testing, data analysis, and fine-tuning to achieve the desired performance characteristics, as it will be described in the next section 1.1.1.

1.1.1 Engine Laboratory Measurements

The development of a new engine product (e.g. engine module, fuel, technology) is based on the virtual creation of the engine (design& virtual validation), the establishment of manufacturing specifications (assembly specifications and supply chain) and finally from the engine laboratory test, which is the backbone of this thesis.

In the engine laboratory a full scale engine, previously pre-assembled in the factory assembly line, is installed and connected to the control room via several system and data connection tools (Figure 2). The engine is equipped with hundreds of sensors, installed on different components, from critical ones as valves to non-critical ones as the engine scale. The sensors are connected to different data acquisition tools (hardware and software) which consent to transform and transfer the signal and the measurement results to the control room and to the data storage [1].

Figure 2. Example of engine under test in Wärtsilä Laboratory.

By testing in the laboratory the engine is checked under different loading conditions in order to verify:

- Engine performances (engine power outcome, fuel consumption etc.);
- Components durability and system robustness;
- Emissions levels (CO_2, NOx, acoustic, vibrations etc.).

Generally the testing is executed based on specific test request, which are internal inquiries planned by project managers and delivered by internal management software adopted in research and development department named “Polarion”. In the test request the description of the overall targets are presented and justified and scheduled by needed due date. Furthermore, the test request contains metadata related to the specific project (commercial/research & development) and information about the personnel and engine involved.

At this point, two separate activity requests are generated from the test request, one for the testing department and one for the engine performances department. the testing department has the duty to complete the assembly of the engine, the preparation for the test and the organization for the test (e.g. consumables), while the performances department has to investigate the correct variables, parameters and factors to test with the selected engine.

A engine tested in laboratory is generally run for weeks or months before substituted by another engine, and each test can be run for 8h/day. Each test request is linked to one test program which is constituted by multiple test series, which are defined by single test points: Table 1 recaps and describes the organization of the test into details. the test program is structured by testing and performances department together, while the test sequences and test points are coordinated by the performances team.

Table 1. Engine Test Description in Company system.

Test Element	Description
Test Request	First step to initiate an engine laboratory test, it contains information about the related project, personnel involved and enables to connect the test to other activities on-going in the company (customer projects or research and development project).
Test Program	The list of test sequences along the entire test slot for the engine. The tests sequences are grouped and organized based on the hardware requirements (e.g. changing of specific components) as well on testing purposes.
Test Sequence	Lists of test points grouped together on the base of design of experiment (DOE) approach.
Test Point	The base of engine test it is defined by specific performance parameters (main headers) which identified specific conditions of the engine. Each test point represents a specific engine testing conditions run by the testing team.

Once defined the test sequences and the test points, the test program is shared with the testing team which runs the engine according to the information described in the program. The test point is constituted by :

- Desired values, data that are decided by the performances engineer and used as input from the testing engineer in the control room (e.g. engine speed, timing etc.)
- Specific measurements results, calculated directly from the sensors measurements (e.g. compressor temperature, fire pressures etc.).

For each test point, the measurements obtained from the sensors installed on the engine are collected and saved in the control room computers, using specific software (named lab tool): the data are divided between slow data and fast data on the base of data acquisition frequency (1 Hz and 1 measurement per 0.5 crank angle degrees~30sec timeframe). The data are then written in excel files called “Prestanda”; each Prestanda describes the Engine conditions, performances and status for the relative test point. Each Prestanda represents an average of 10 minutes of test run.

The Prestanda files are stored in SQLITE database and then synchronized with Azure files database. The Prestanda files are read by specific software (OneCalc) which allows the performances engineers to analyze the data stored in all the excel files and review the engine performances.

1.2 Laboratory Measurement Database Project

1.2.1 Problem Statement

The data acquisition model described in the previous chapter, described a robust methodology to obtain and conserve the data from engine measurements, however it contains some uncertainties in its workflow.

The measurements results stored in the database represents only average values, while the precise results are removed from the database because of data storage limitations. In the case of combustion process, the precise results are highly important and needed for combustion and engine experts, particularly when simulations are involved. Virtual validation of engines by simulation requires solid references to be used as boundary conditions.

Data are formatted by proprietary software companies and the users need to purchase and keep an extra license to operate with the data. This requires extra costs and creates risks for supply chain blocks and uncertainties, such in the case of supplier inconsistencies.

The measurement data cannot be accessed at once since there are poor metadata linked to the measurements results. Indeed the Prestanda files contains partially some reference metadata (e.g. test ID and date) but they cannot be reached easily by company experts except performances experts. The absence of simple research makes the research of data for simulation and engine development long, imprecise and risky since it is done manually.

The traceability and the quality of the data is compromised from the loss of information along the process, particularly the information related to the hardware used in the laboratory engine test room, such as sensors serial number, which can influence the final measurement value. Indeed it is highly frequent that incorrect calibration and sensors failure gave incorrect results or completely non-sense measurement results. In the last case Performance Engineers are able to recognize the error based on experience and knowhow, but in the first case, the incorrect measurement could be very small and scale the entire final value (such as oil consumption) to un-expected and unreachable mistakes.

Improvements in the data collection and quality could lead to better usage of engine measurements results and could lead to possible implementation of Machine Learning solutions. Indeed, considering the complexity of engine performances and the amount of data collected from each test point and test program, the implementation of machine learning in the engine development has been frequently proposed in Wärtsilä. However, one of the requirements for the correct implementation of machine learning stands on the quality of input data as well as in the full traceability of them [2]. As showed in Figure 3 and described by Sculley et all. [3] in their proceeding, the machine learning system is constituted by different aspects, mostly related to the data (collection, verification, maintenance and monitoring) rather than the coding itself and the tools used to create the machine learning system.

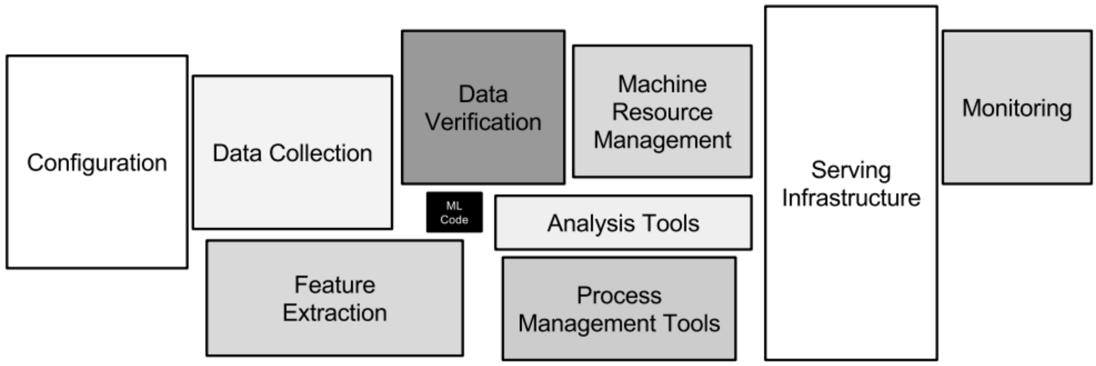

Figure 3. The hidden feasibility problem of introducing machine learning inside companies **[4]**.

1.2.2 Project Plan and Purpose of Thesis Work

Based on the problems described above in Chapter 1.2.1, Wärtsilä has approved a project to create more robust data acquisition and storage solutions for its laboratory engines; basically the establishment of more accurate laboratory measurement database, connected not only to measurements results obtained from the laboratory, but also all the metadata related to the tests and measurements. The key factor in this project is to structure the data using specific database standards like ASAM ODS (Open Data Services) or Open MDM (Described in Chapter 2.3), and create an application solution for the customer interface, specifically a web browser portal able to navigate in the new database. Figure 4 shows a schematic representation of the database skeleton and the different sources of the information collected into the database: indeed, the measurements results come from the engine laboratory as described previously in the introduction section (Chapter 1.1.1), but differently from before will be linked also with the metadata of the data acquisition tools (hardware and software) and the metadata coming from other company programs and database such as Polarion (test requests information) and company PLM system. Based on python coding then, the database is created and together with this also the end-user interface.

The establishment of such database and method behind will create several advantages like:

- Enable improved traceability of data and metadata with the measurement tests, facilitating the data monitoring;
- Facilitate the end-user access to measurement data, facilitating simulations and virtual engine simulations;
- Create the base element for trustable and precise machine learning system for engine performances and future engines development.

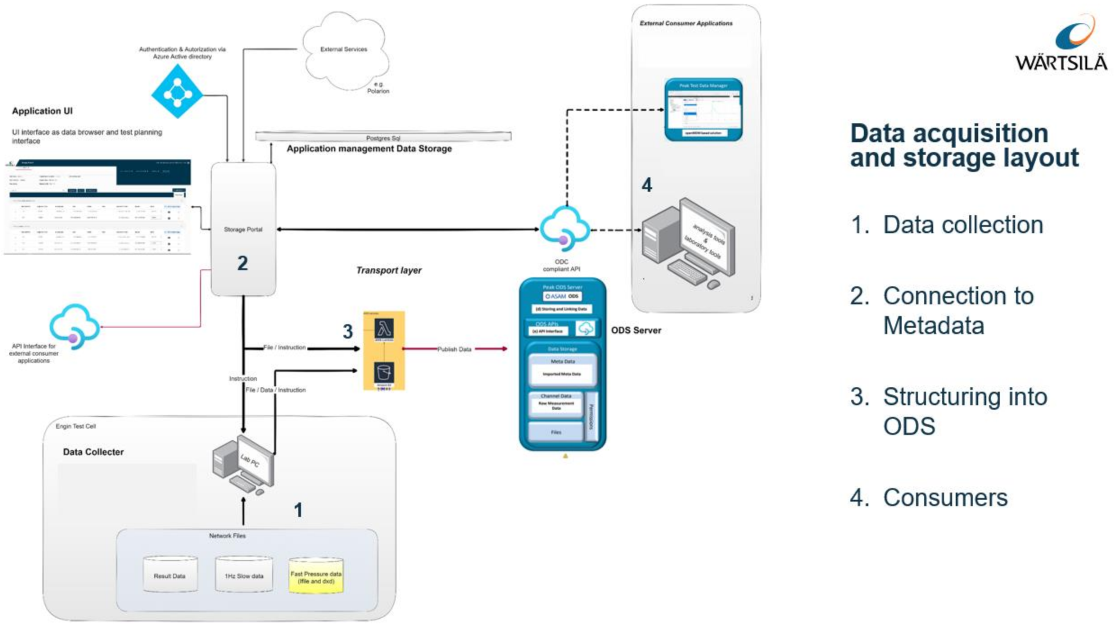

Figure 4. Schematic representation of the Laboratory Measurement Database Skeleton.

The base of this project stands on the structure the database should have, particularly how the information between data and metadata will be linked together. This is the main topic of this thesis work, the creation of the UML map of the laboratory measurement database. This structure will be used as guideline and scheme on which creates the coding and applications which will establish the database.

2 Materials and Methods

2.1 Thesis Objective: Database UML Map

The objective of the thesis has been identified as the creation of UML map for the engine laboratory, specifically one engine cell present in Wärtsilä (Engine cell number 2). The work in the thesis has been based on collecting all the information needed to create such UML, Unified Modeling Language, map, adopting a Class Model so defining the main classes, attributes and relationship which builds the database [5].

The class is the defined as the basic logical element in the UML and determines both the data and the structure of the specific unit. A Class stands as a template which instances are created at run time. As described in Figure 5, each class model is defined by Class Attributes, which identify the specific Class and the Class Operations which describes the behavior. [6, 7]

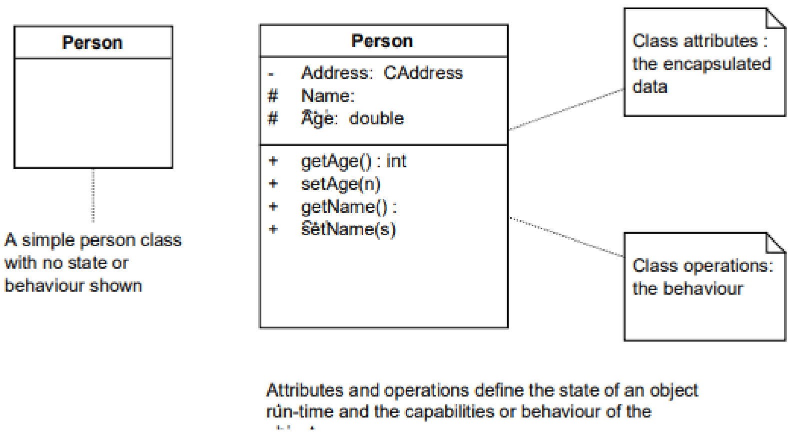

Figure 5. Example of Classes, attributes and operations **[6]**.

Moreover, different Classes can be linked by relationship, which identify the structure of the information [6], for example:

- Association, where one class have access to read/manipulate attributes of another classes;
- Aggregation, where a collection of different classes are included in another class; Composition is a strong form or aggregation, for which an actual Class is composed of other classes (Figure 6).

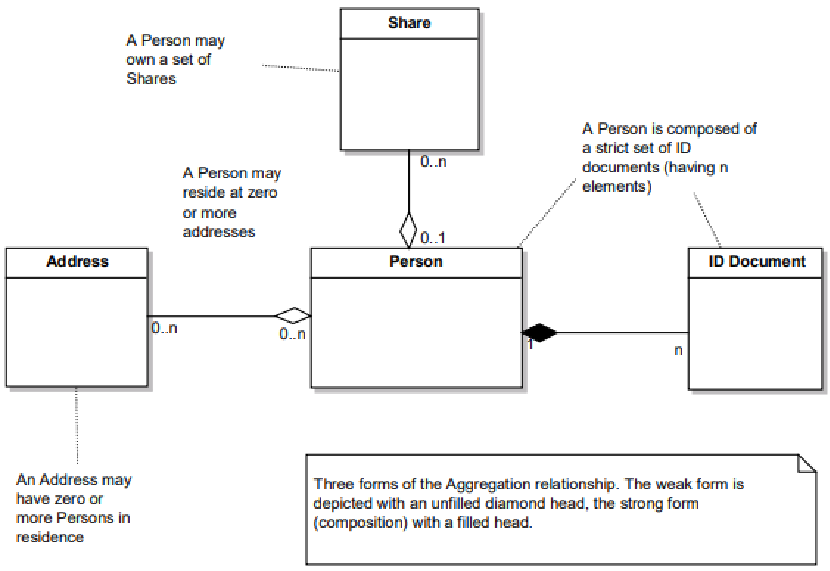

Figure 6. Example of Aggregation relationship **[6]**.

The classes and attributes needed to describe the database have been obtained mostly by Interviewing internal Wärtsilä Colleagues as described in the next Chapter 2.2. The UML Map has been created using UML Star software.

2.2 Interviews

The main classes to be investigated for the database are described in Table 2 and indicated by ASAM ODS standard [8], however there are many others classes that are linked to the principal three ones and collects metadata important for the analysis (e.g. measurements and data information). These separate classes will be described in the final UML map description (Chapter 3.1).

Table 2. Core UML classes for the Laboratory Measurement Database.

Database Element	**Description**	**Dedicated Team**
Unit Under Test	The engine under test in the Test Cell	Laboratory Testing Team
Test Sequence	The information specified in the Test program.	Performance Team
Test Equipment	The data acquisition tools, both hardware and software, for all the measurements.	Test Equipment Team

The three main areas analyzed for the database represents the core of the measurement database and the key area of the laboratory and that is why the thesis work has been focusing mostly on describing these elements as using class model, particularly identifying the principal attributes for each class.

2.2.1 Interview structure

The interviewing work started initially by identifying the responsible managers for the respective teams (Table 2) to request the time allocation from their team members and to identify the respective experts of the teams who shall have been contacted for this work.

Since each team is responsible for different knowhow, from engine combustion performances and engine testing to instrumentation hardware and electronics, the interview results differed from team to team, however the same interview methodology (Figure 7) has been used for all of them.

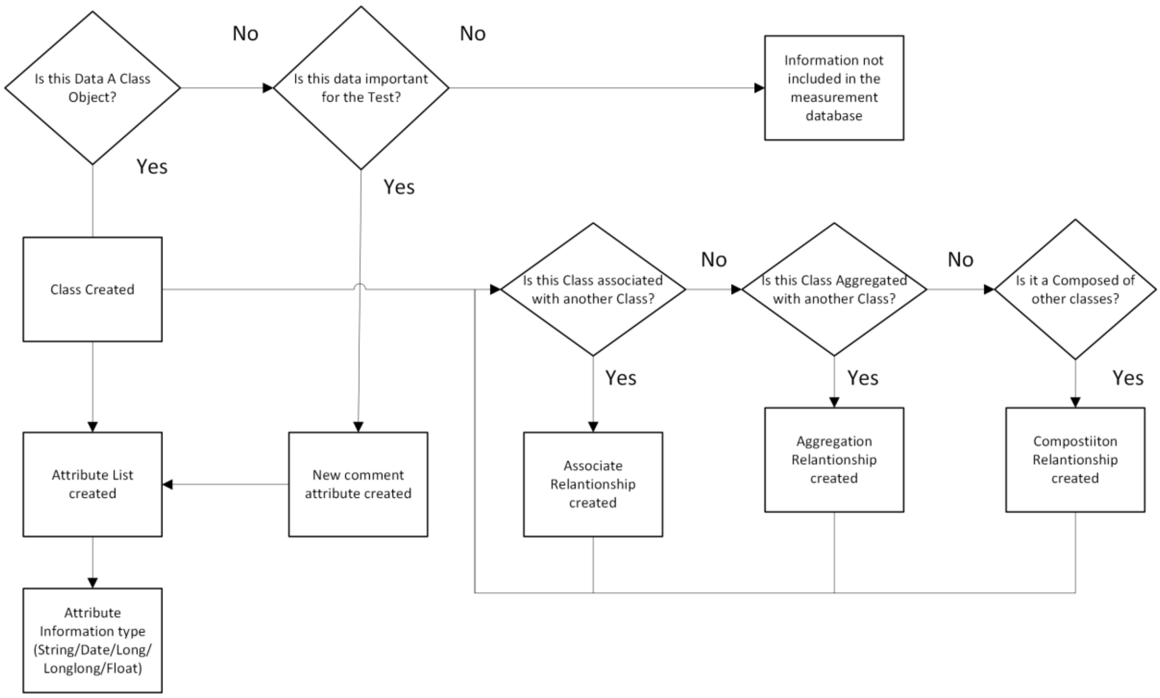

Figure 7. Workflow adopted to interview the Experts Teams in Performance, Testing Equipment and Engine Laboratory Testing.

By adopting the same method, it has been possible to focus on the nature of the data and the usage of it particularly; indeed the most difficult analysis during the interviews, has been to scrutinize the information received to filter out the information which were not relevant for the database and the scope of the project.

For example, the supplier of a testing equipment like a sensor is an important information to keep for the instrumentation, but it is not relevant for the testing purposes or traceability. Indeed all the supplier information, so all the attributes related to the class supplier, are stored in a separate database (SAP) and can be retrieve using the attribute serial number of the sensor class. So the crucial information to store in the measurement database are not the supplier class and attributes but only the class sensor and the attribute serial number, which creates the link with the corresponding supplier in the SAP database. By this approach, the database becomes more slim and light and the user is able to access only to relevant data.

The collections of responses from the Target Teams has requested not only to understand the type of classes and attributes presents in each core area (unit under test, test equipment and test sequence) but also to understand the detailed way of working for the testing and more specifically in each team. Furthermore, another important aspect of the thesis has been understanding how the interviewed teams would have used the new database and future application, based on the UML map. Indeed the map itself describes the structure of the database and also the way to navigate through it.

In the next Chapter the standards used for the database, ASAM ODS and Open MDM will be introduced and described, describing also the solution found to facilitate the navigation through the database (Mapping o Data).

2.3 Database standards

2.3.1 ASAM ODS

The automotive industry has been working on the standardization of data management since many years, indeed from 1998 the Association for Standardization of Automation and Measuring Systems has been working on the standardization of Measurements Data, since the very different approaches used by the players in the market [9] [10]. Indeed as described in Chapter 1.2.2, having an organized data structure, possibly with standardized interfaces, minimized the effort for system integrations with heterogeneous environments and new testing conditions.

The ASAM ODS (Open Data Services) standard focuses on persistent information storage and retrieval for the measurement data storage and maintenance [11] as schematically showed in Figure 8. Specifically, ODS represents a lightweight integrated collection of data associated with current-valued information from the different data origins of up-to-date operational data. The main difference between a standard data warehouse and the ODS lays on the volatility of data: in the first case, data non-volatile and fixed, while in the second case they are not meaning also that the timeliness of data in the system is much more flexible differently from the first scenario [12]. By Using ODS database it is possible to extracts, cleans, transforms, filters and integrates data generated by various application systems and provides data sharing services like real-time cross-system operation reports. ODS also supports collaborative applications by utilizing cross-system operational data and related analysis results. [13]

Generally, ASAM ODS prescribes:

- A standardized data model, base model, to avoid ambiguous definition of data and to provide a rough classification in order to interpreter similar data from different applications and systems;
- Standardized interfaces for data storage and retrieval, so to operate on different ODS data sources and Standardized text-based formats for data exchange;

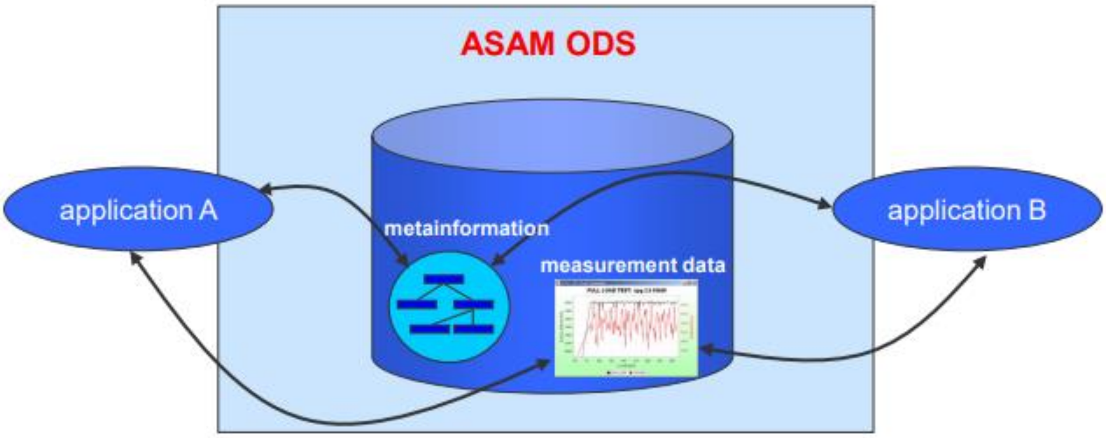

Figure 8. Schematic Representation of ASAM ODS standards usage. **[8]**

ASAM ODS standard is mainly used in test automation and test bed systems, like for example: test data management, measurement & calibration, integration of automation and measurement systems, simulation, data post-processing etc. [9] [14].

Within this thesis a useful branch of ASAM ODS, has been the Associated Standard of Instrumentation, which focuses specifically on the data acquisition tools, their configurations for particular measurements. This specification in the standard, provides rules for the application elements used for documenting the instrumentation, specifying the attributes and the relationship between each other, as well it prescribes the rules to create instances of these application elements [8].

Examples of associate standard of instrumentation are showed in Figure 9, showing a general data model and UML map for measurement database system and Figure 10 focusing on the test equipment data model.

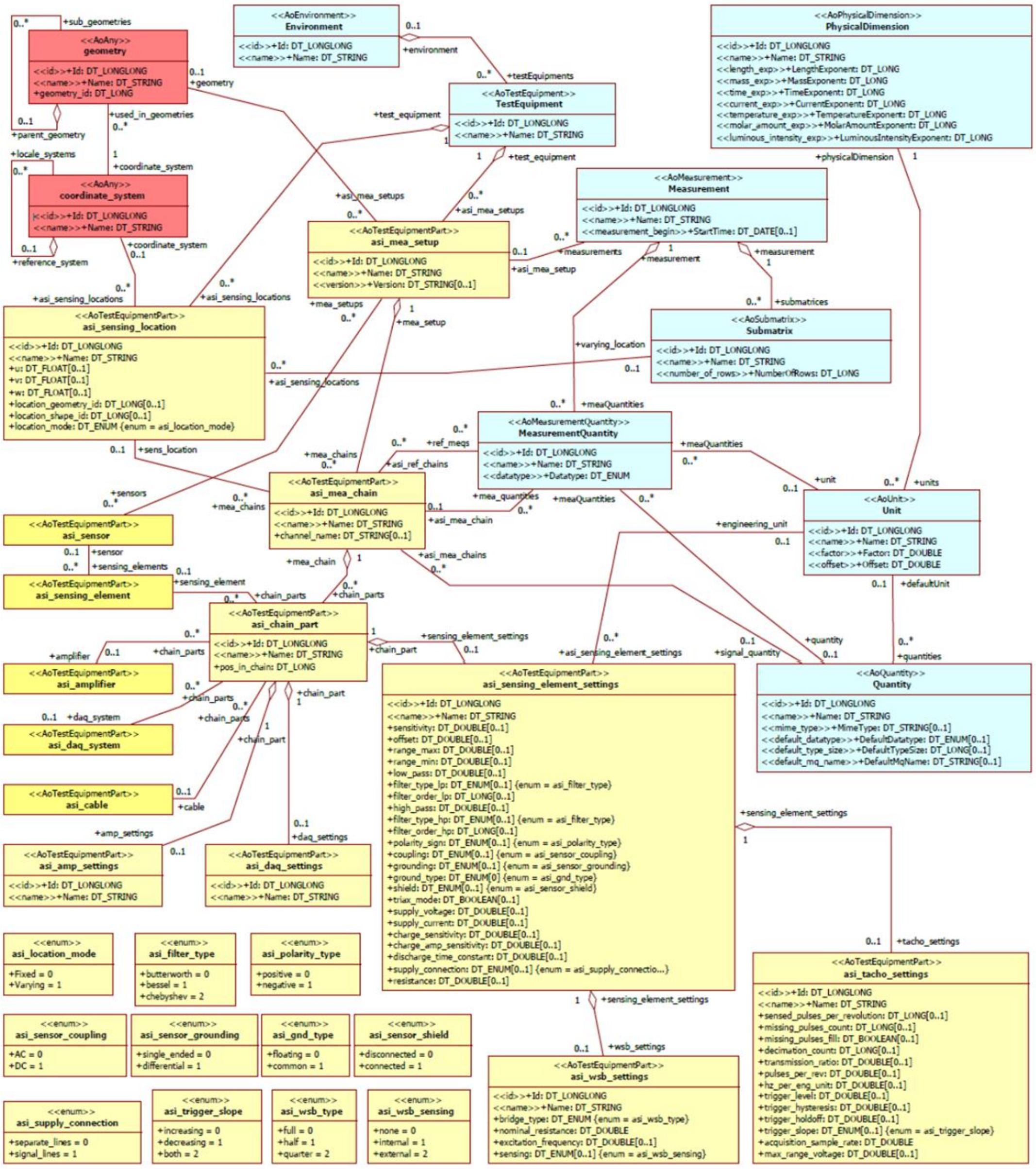

Figure 9. Example of data model for Measurement Results Database **[8]**.

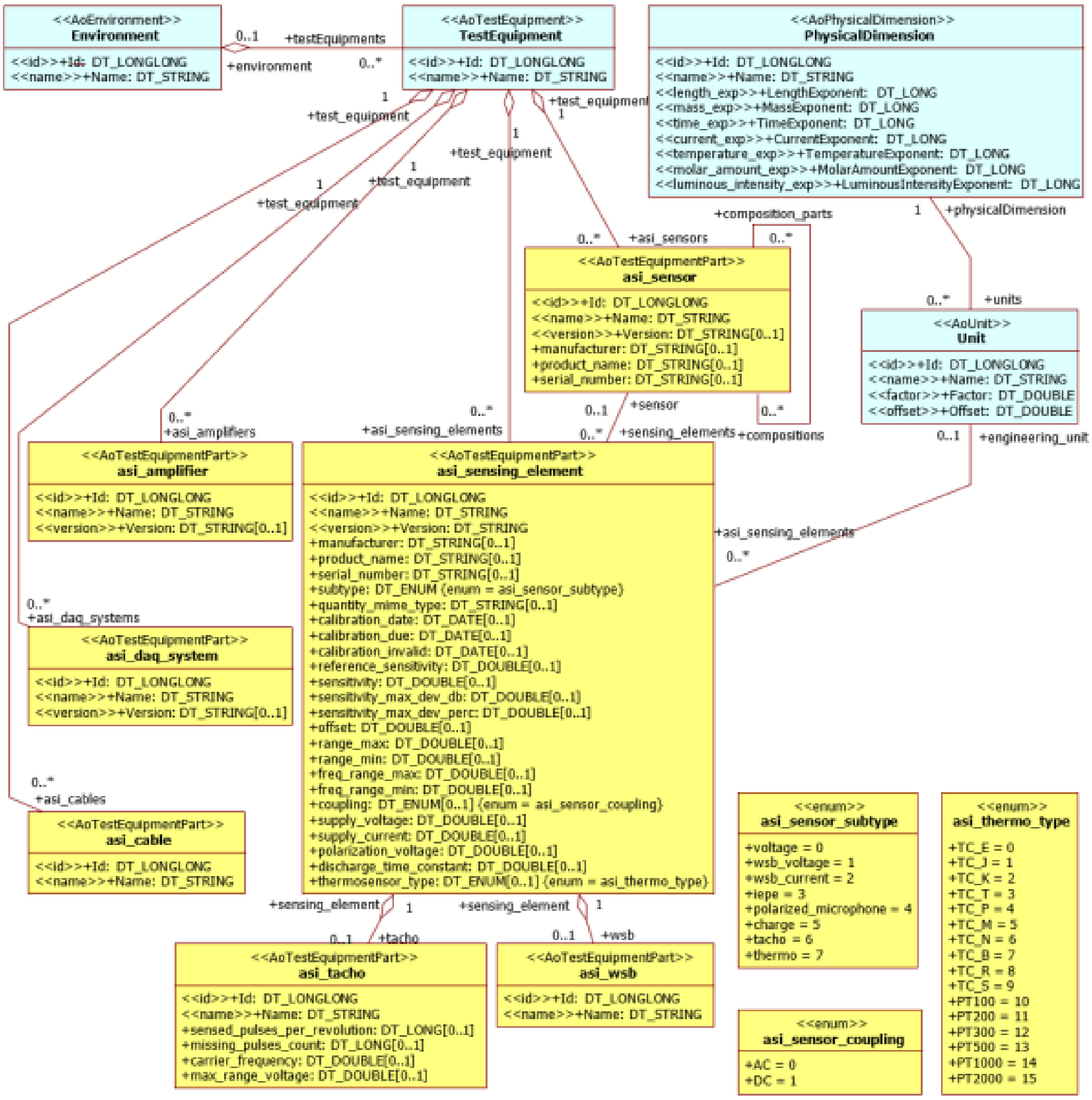

Figure 10. Example of associate Standard of Instrumentation mainly focusing on Test Equipment. **[8]**

As described in the pictures, the data model is described by UML class object model where the classes are divided in to separate groups according to color scale and naming procedures: measurements (and their subclasses), test environment and test equipment in light blue while all the test equipment part and subclasses in yellow. Almost all the classes in these data model are described by specific list of attributes which can work as example or template for other industrial applications of ASAM ODS as in the case of Wärtsilä Testing laboratories. A peculiar element present in the associate standard of instrumentation, visible in Figure 9, are the data mapping classes called specifically ASI Measurement Chain (ASI Meas Chain in the model) and ASI Chain Part. These classes represent chains of data and facilitate the management of metadata from Instrumentation.

Each class in the data model, which represent an actual existing element (hardware or measurement) of the database is defined by an ID and Name, for example ID:1234 and Name Sensor. This sensor is linked to different information, firstly its own subclasses (e.g. Sensor settings) as well to several different measurements data. Without the use of a chain, in case of a sensor change, (e.g. ID1235, Name Sensor1), the entire list of linked elements (e.g. sensor settings) will require a new ID code itself, since their linked element changed. As a results, the amount of data will start to increase dramatically if scaled to all the elements of the test equipment (Sensor, Data acquisition tool etc.). Instead, thanks to the use of Chains, the effect of changing one element in the chain, update only the ID of the chain, creating one single new element, while all the rest of classes connected to the chain can maintain their own ID and they are not duplicated. Such peculiar solution has been helpful and fundamental in the thesis work, since for most of the development of the UML map, the acknowledge of such element was not evaluated.

2.3.2 Open-MDM

Another application model adopted in the development of the UML, has been the Open MDM, a generic application model based on ASAM ODS standard principally applied on the automotive industry [11]. Differently from standard ASAM ODS, which includes specifically data storage and data transfer, Open MDM included also the test process and the test preparation, as required in the case of measurement laboratory database project. Indeed such approach facilitates the design of experiments inside companies, because the system can be also integrated with improved navigator and search function, as well as other simple output features such as chart view and export functions [11].

3 Results and Discussions

3.1 Final UML Map

The UML map of the database is graphically showed in Figure 11 and it will be described in details in this section. Indeed, as highlighted in Figure 12, the entire data model can be divided in operational groups based on the main targeted data source/target they are describing, specifically:

- Test Request Data;
- Measurement Data;
- Test Sequence Data;
- Unit Under Test Data;
- Test Equipment Data.

In the next subchapters, each of this subgroups will be described. Below in the Figure 13, the overall legenda for the class relationships is highlighted: these color rules have been fundamental in the description of classes associations and aggregations since in the UML software used (Free Software StarUML) it was not possible to indicate the father-son relationship, in the class description (operation field). For this reason, the relationships have been added to each class, and the hierarchy of information has been described by color scale.

As showed in Figure 13, the classes can be linked based on the information they are carrying (info) or based on their hierarchy (father-child), so if one class is derived or not from another class. The distinction between mandatory and optional, depends on the final user interface, meaning how that information will be used or not in the final web application. Indeed, if a relationship is mandatory the user will always have to consider them in the data results browser/search, while in the case of optional information, the relationship exist in order to full fill the traceability requirements.

Figure 11. Overall representation of UML map for Measurement Laboratory Database.

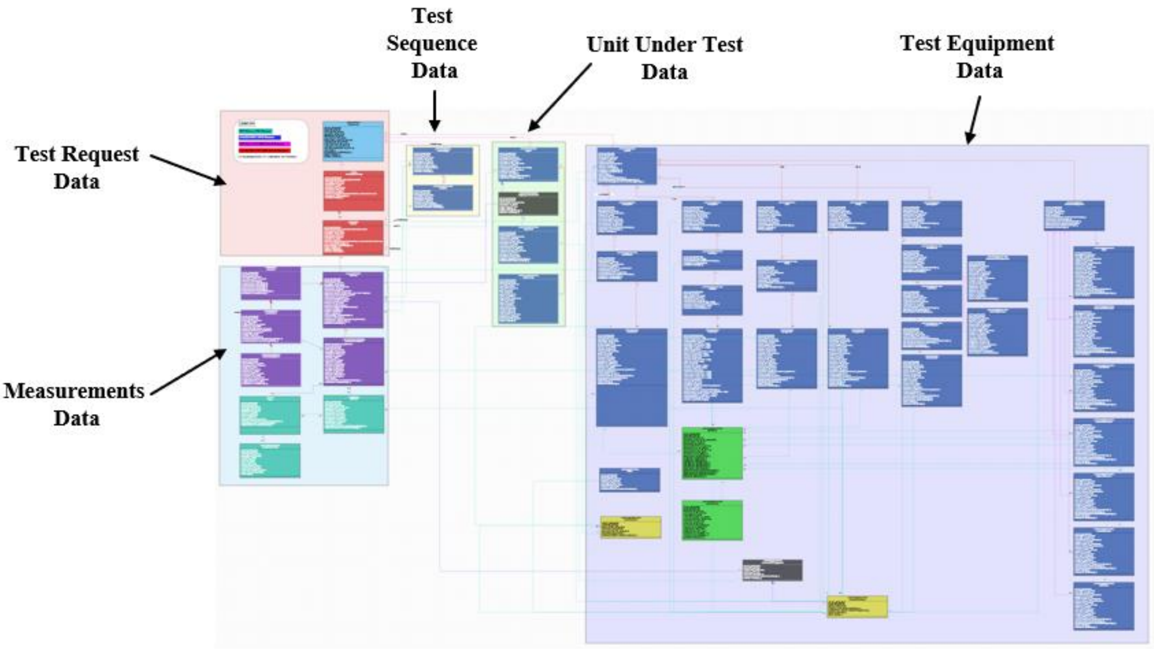

Figure 12. Graphical division of the data model classes of the database into functional groups.

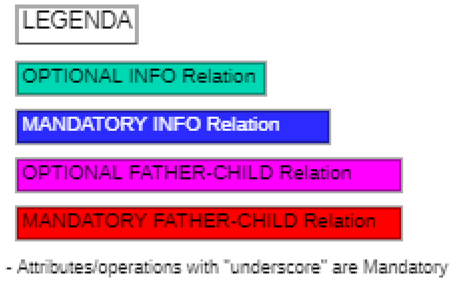

Figure 13. UML Map Legenda.

3.1.1 Test Request Data

Highlighted in Figure 14, the Test request data starts describing the Enviroment of the database (the measurement results space and data model), which represents the highest element in the hierarchy of the data model: it describes the space where the data and the requests are collected and stored: indeed in more details, it can be seen that is aggregated of the test equipment (”TestCell”), the unit under test (”Egine”) and test sequence (”TestCampaign”). These classes, as well the subclasses are all included in this single entity, and that is why the aggregation relantionship (as illustrated in the picture). Moreover, it can be observed the two classes representing the test request, which acts as links to the external database where the metadata are stored.

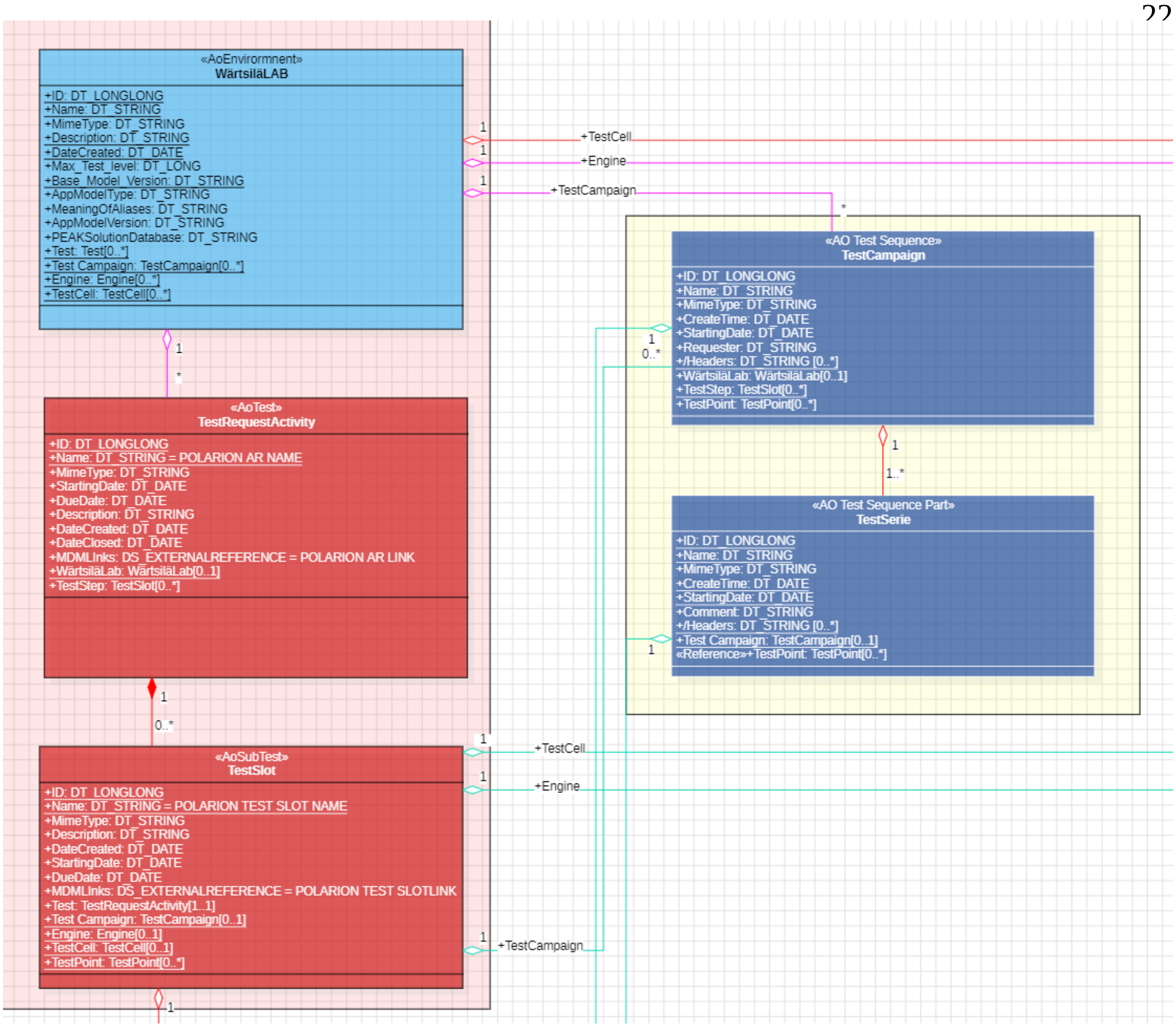

Figure 14. Test Request Data on the left and Test Sequence Data on the right.

3.1.2 Measurement Data

Figure 15 shows the classes related to the measurement data and their storage, basically the core of the database in terms of information. All the measurements results, from slow data and fast data (described in Chapter 1.1.1), are described using these classes: the calculated data, based on the measurement data, and possibly useful for the performance engineers are left out of these classes. Indeed their calculation can be operated by the final user application, saving space in the database.

The most important class in the group is the TestPoint one, which specifically describes the input described by the performance engineer during the test sequence and test campaign (exactly the headers attribute in the test point class). By defining these, the user will be able to navigate through the database on the base of the specific headers and obtain the related measurement results as well as the possible information over the test sequence, unit under test and test equipment used.

Aggregated to this class, there is the MeasurementColumn, which represent the actual list of measurements results obtained for each TestPoint in the TestSequence. This class is then directly connected to the TestEquipment through the MeasurementChain. One important attribute of the MeasurementColumn, MeasurementChain and also Sensors (described in the TestEquipment) is the Channel. The Channel, identified by specific ID, is generated automatically by instrumentation team when they instrument a sensor on the UnitUnderTest.

The Channel ID identify one specific sensor and also identify one specific measurement results, so it represents the link between the two separate elements and it allows the traceability of the data. The use of the measurement chain enables higher safety in the reliability of this link, since it enables to record the malfunctions and errors of mounted sensors: in case a sensor is broken and substituted, the channel won´t change name and ID, even if the sensor will be different (e.g. serial number). By using the measurement chain instead it is possible to generate a new ID for the measurement chain which conserve the link to both sensor and measurement value. More description of this class will be addressed in the Chapter 3.1.5.

This is specifically the direct advantage of establishing a measurement database based on class object data model and class relationships. The performance engineer, or whoever will access the database, will easily navigate through terabytes of measurements data with customized parameters/information research tool. Indeed, nowadays the research is done by the performance engineers who knows in which folder the Prestanda file are saved and can read them by specific software: without the performance engineer the research of the values is not practical for the other experts in the company.

Last elements of the measurements data group are the classes specific for units, quantity and physical dimensions, as showed in green in Figure 15. This information are mandatory for the database and are described based on the ASAM ODS Template (Figure 9).

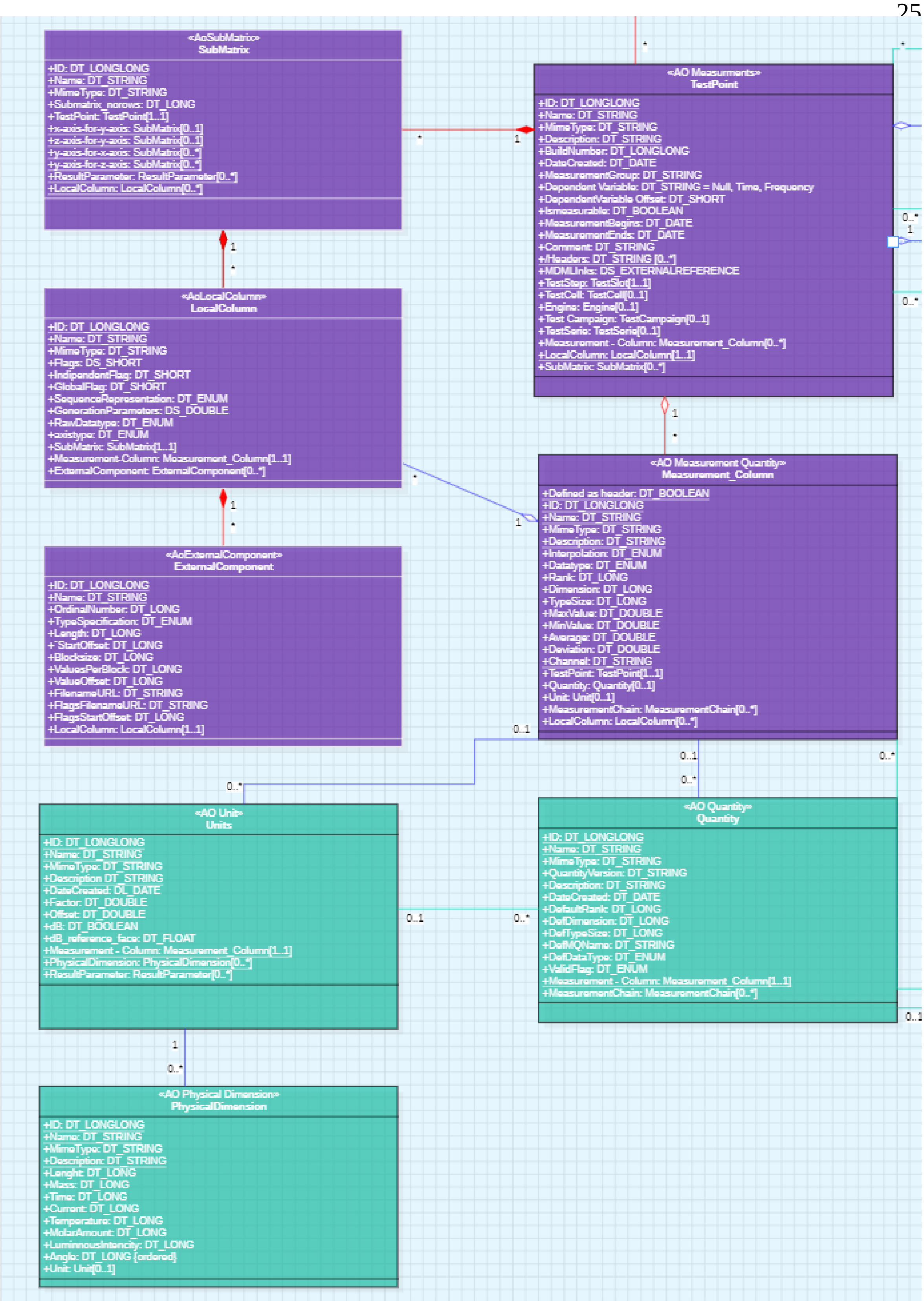

Figure 15. Measurements Data Classes and their relatives subclasses, representative of the physical information (green boxes).

3.1.3 Test Sequence Data

Described in Figure 14, the two classes, TestCampaign and TestSequences describe the upper level of the Test Planning and are principally linked to the TestRequest as well with the TestPoint classese, since it represents the lowest level of Test Planning. Comment attribute is an important feature of these classes since it allows to collect Testing Engineers and Performance Engineers comments, previously lost in the Performance Engineers excels.

3.1.4 Unit Under Test Data

The UnitUnderTest represents the engine under testing in the TestCell so its attributes must be able to describe it precisely, helping also the user of the database to filter the list of tested engine in a simple way. Indeed, as showed in Figure 16, the engine is identified not only by its serial number and build number (number assigned for the assembly phase), but also by the main fuel adopted by it as well the turbocharger. Each engine is physically constituted of hundreds of engine parts, all listed electronically in the bill of material. This list represents the engine in the moment it exit from the assembly line and reach the test cell and it does not change for production engines. Although, during laboratory engine tests, the engines might be subjected to modifications, part substitutions, part redesign so the list of components changes.

In order to avoid duplication of data, in the case of engine parts the Mapping Class (Figure 16) has been introduced: basically it acts as bill of material which contains the array (list) of components for the same engine build number. In this way, if the array changes because of a substitution of a part, the Mapping ID will change and will possess a new array of engine parts but it will still refer to the same Engine, so same original Build number. Without the mapping for each components replaced, or removed from the engine, a new Engine class would have been created, so all the existing engine parts in that engine would have been duplicated in the database. Mapping solution has helped to avoid a huge computational cost on the database and has made the data model smarter.

Connected to the topic of substitution, in the UnitUnderTest group the EngineDiary class has also been included, since it represents the list of operations and comments accumulated during the engine test. Each attribute of the class represents an actual information currently registered in the laboratory engine room software (Labtool) and it gives a complete history of the testing of the engine, from components parts and testing point of view. Indeed one mandatory attribute of the EngineDiary is the OTRunhours, which is an actual counting of the hours from the beginning of the engine test: for each part installed or substitutes, this attributes is showed, and describes at which point of the engine test the Testing Engineer has done modifications or comments.

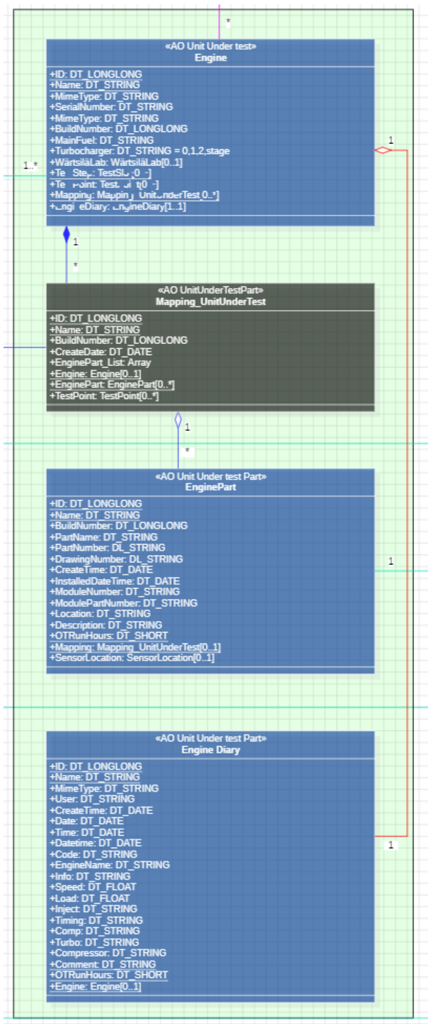

Figure 16. Unit Under Test class group.

3.1.5 Test Equipment Data

The widest part of the UML Map is constituted by the TestEquipment group because for the complexity and different systems present in the engine test cell. The highest class in the hierarchy of this group is the TestCell itself: it connects directly with all the main classes described in the previous chapters as Figure 17 illustrates. Aggregated to it there are all the different DataAcquisitionTool Chains (Figure 18):

- UNIC System, the automation system installed on the engine connected to sensors which are part of the engine;
- PLC System, the chain of sensors connected to the PLC system of the test cell;
- SDAQ, Single purpose Digital Acquisition system connect to the LOG (Laboratory OPC UA Gateway);
- MDAQ, 8-channel Multipurpose Digital Acquisition system, still present in the Test Cell 2 it will be completely substitute in the future by SDAQ;
- DEWE System, data acquisition tool specific for fast data;
- Special Sensors, list of sensors connected to LabTool and Test Control Room.

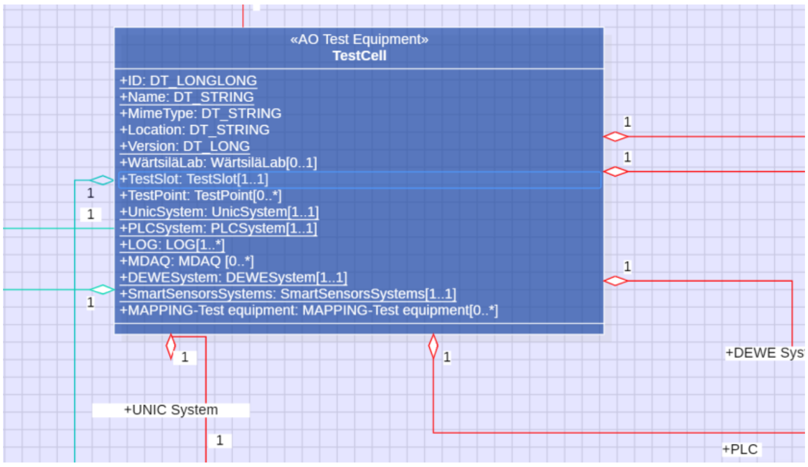

Figure 17. Test Cell class in Test Equipment Data group.

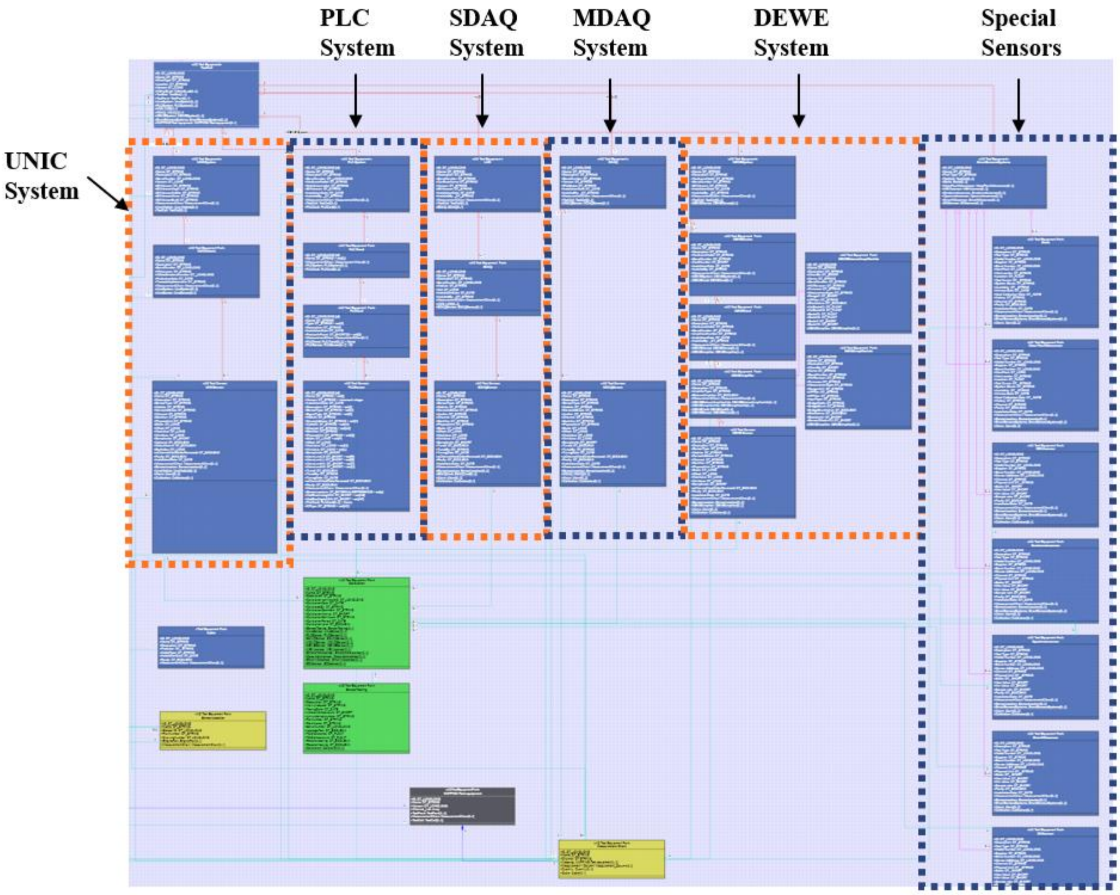

Figure 18. Schematic representation of the Test Equipment Parts, divided into the different Data Acquisition tool.

The structure of the data acquistion group is described in Table 3 and illustrated in Figure 19.

Table 3. Structure of the Data Acquisition Tools in the Test Equipment data of the UML Map.

Structure Level	**Structure Element**	**Description**
Highest Level	Data Acquisition System hardware	It represents the actual system, hardware and software, which collects all the information coming from the lower levels and delivers them to Labtool software in test Cell control room; In the case of Special Sensors, this level is missing.

Intermediate Level	Data Acquisition System Sub-assembly in the hardware	It collects all the intermediate systems, hardware, which constitute the data acquisition system; the list of intermediate systems can be empty as in the case of MDAQ, or filled with different elements as in the case of DEWE systems; the main function is to collect and transfer the information from the sensor to the higher level.
Lowest Level	Data Acquisition System related Sensors	The sensors are the physical elements collecting the information from the engines; they are divided into separate groups according to the specific system they refer to; they are the lowest level of the structure meaning that they are not constituted by any further class.

As mentioned above, each data acquistion tool has been considered as chain of systems as done for the unit under test: in this way it is possible to limit the duplicates of data inside the database for each data update in the element of the chain. For example, in the case a card of dewe system is substituted (intermediate level in the chain) there would be the need to duplicate all the information regarding the amplifiers and sensors referring to the new card, even if these have not been modified. By using the chain approach instead, the ID number is updated and the link information is updated without multiplying the information.

Figure 19. Detailed view of Data Acquisition Tools classes in the Data Model.

Connected to these chain there is the actual MeasurementChain and related Mapping, plus the sensor location, as described by ASAM OTS and shown in Figure 20. Sensor location allows to connect each specific sensor to the related UnitUnderTestPart on the engine, while the Mapping, aggregated with the MeasurementChain (Child) allows to connect the sensors with the related Measurements thanks to the Channel attribute and the relational link. Thanks to these database tools, the complete traceability of the measurements would be established.

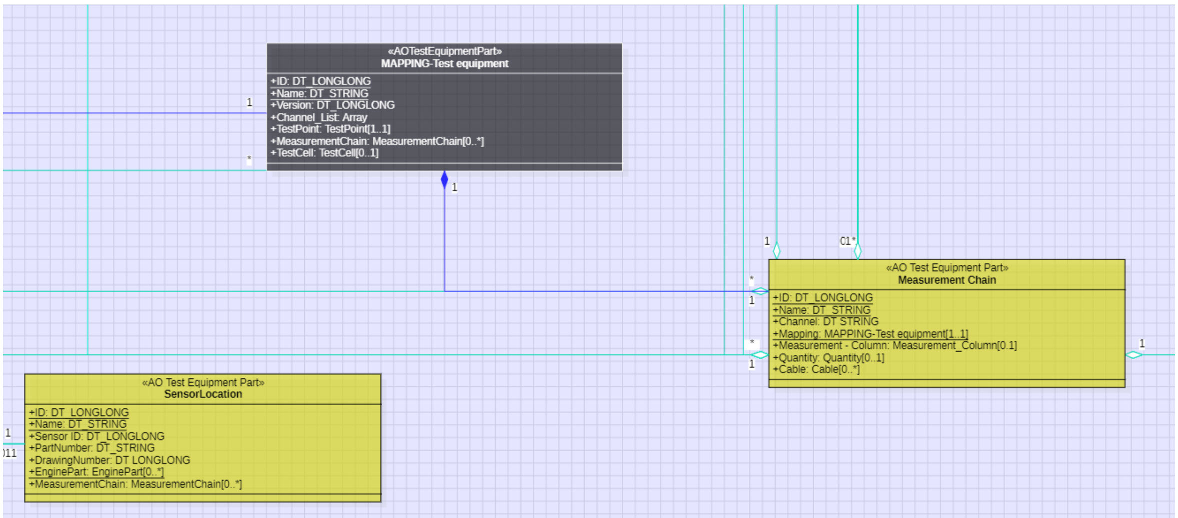

Figure 20. Focused view on TestEquipment Chains as described by ASAM ODS.

The final elements of the database, as shown in Figure 18 are the class related to Cables, the Sensors Calibration ones. In these classes the main attributes connected to sensors are described and add additional information to the instrumentation data traceability. Indeed particularly in the case of calibration, the incorrect scaling factor or sensor calibration could lead to noticeable or not noticeable misread on the engine performances, from negative combustion temperatures (noticeable error) to slightly excessive fuel consumption (not noticeable error). These classes are integrated in the database particularly for this reason, being able to retrieve all the information in case of instrumentation faults or misread.

4 Conclusions and Future Work

The creation of the data model and UML for the new Laboratory Measurement database has been described highlighting the solutions and the impact of the choices adopted in the construction of the database map. By implementing ASAM ODS standards on Wärtsilä Engine Testing procedures and data storage, it has been possible to create a smarter and lighter database, in which the user can navigate more easily than existing database and with reduced amount of data. The usage of standardized templates for the creation of the data model has been the seed for the creation of the UML map itself, however the largest effort has been taken in interviewing the internal teams involved into the project. Understanding the current way-of-working and procedures adopted in the measurements data storage, instrumentation and testing equipment created the base for the data model and required tens of hours of interviews, meetings and discussions with Wärtsilä experts.

The consequences of establishing an ASAM ODS based data model and database have been the driving factor for the laboratory measurement project inside the company and for this work. Indeed, having a easier-to-access database and more data linked together with less overall dimensions will enable a more accessible use of the engine performance results as well the utilization of real data measurements values into simulations and machine learning applications. At the same time it will allow more experts from the company to access the data for their study and system improvements on the engine and will enable more accurate traceability of the information for testing, unit under test and instrumentation. This requirement will become more and more important in the next years, particularly for class societies, so the data model has been considering it in its structure.

A similar reference case can be found in Bosch Companies for their laboratory data management [9]. Similar challenges to Wärtsilä case, the German company adopted ASAM ODS database for their laboratory management solution which cover hundreds of test cells worldwide: many laboratories have developed their own data storage solution and standards, duplicating efforts and costs and widening the complexity of data traceability. The solution standardized test documentation and offered time-saving functionalities like search and improving the productivity of test engineers. The hardware investment for storage of measurements was reduced through a centralized approach. [9]

The data model built in this thesis will act as foundation for the Laboratory Measurement Project and together with the UML map it will be used by IT professionals as guiding map for the creation of the user interface for the database [15]. Once completed the database will be tested in test Cell number 2 in Wärtsilä premise and then proposed as solution also for the other tests cells, included the production engine ones. The strategic target of this project is to optimize the understanding of engine performances and measurement results, by revolutionizing the collection, storage and management of data.

Once deployed, future development and studies will be then taken on the implementation of machine learning solutions to analyze and utilize the data stored in the database. The possible applications of machine learning could vary from engine performance optimization during engine test campaign development, to predictive maintenance of the components installed, to the creation of virtual laboratory engine test, where a virtual engine is ran for several tens or hundreds of simulations based on the measurement data. Machine learning indeed could open up various powerful opportunities for research and development of complex engine systems, but it would only work if the used data would be clear, trustful and organized [16] [17].

5 References

[1] P. Valkjärvi, "Integrated Data Acquisition for State-of-the-Art large bore engine test cell," Vaasa University - School of Technology and Innovations, Vaasa, 2022.

[2] A. Jain, H. Patel, L. Nagalapatti, N. Gupta, S. Mehta and S. Guttula, "Overview and Importance of Data Quality for Machine Learning," *KDD,* no. August, pp. 23-27, 2020.

[3] G. H. Scullery, D. Golovin, E. Davydov, T. Philips, D. Ebner, V. Chaudhary, M. Young, J. Crespo and D. Dennison, "Hidden technical debt in machine learning systems," in *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015*, Montreal, Quebec, Canada, 2015.

[4] J. Lin and D. Ryaboy, "Scaling big data mining infrastructure: the twitter experience.," *ACM SIGKDD Explorations Newsletter,* vol. 14, no. 2, pp. 6-19, 2013.

[5] S. W. Ambler, "Mapping Objects to Relational Database," 1999.

[6] G. Sparks, "Database Modelling in UML," Methods & Tools e-newsletter, 2001.

[7] Y. Ou, "On Mapping Between UML and Entity-Relationship Model," in *The Unified Modeling Language. Physica-Verlag HD*, 1998.

[8] H. Beckers, "ASAM ODS V6.1.1_Release Presentation," ASAM-Association for Standardization of Automation and Measuring Systems, Aachen, 2021.

[9] ASAM, "ASAM Solution Guide," Association for Standardization of Automation and Management, 2017.

[10] J. Feng and J. Hobelsberger, "ASAM-ODS: An Open Standards Engineering Database for NVH Activities,," *SAE Technical Paper,* vol. 01, no. 1441, 2003.

[11] J. R. Schönwald, C. Forsteneichner, D. Vahrenhorst and K. Paetzold, "Improvement of collaboration between testing and simulation department on the example of a motorcylce manufacturer," in *International conference on enineering design, ICED19*, Delft, 2019.

[12] C. Ramakrishna, P. Dr.Srinivas and K. Krishnendu, "Empowering Next Generation Flexible Operational Data Stores with Service," in *International Conference on Next Generation Web Services Practices*, 2005.

[13] C. Zheng, X. Huang and L. Huang, "An Improved ODS Platform on Large Enterprise Level," in *International Conference on Mechatronics, Electronic, Industrial and Control Engineering*, 2015.

[14] J. Wachtler, "Design and Implementation of parameterizable data import for the jvalue ODS," Friedrich-Alexander-Universit¨at Erlangen-Nurnberg - Department Informatik, Nurnberg, 2020.

[15] H. Mohammed, "Storage Portal: A React-Redux Web Application Utilizing Polarion for Test Data," Vaasan University of Applied Science - Information Technology, Vaasa, 2020.

[16] S. Posch, H. Winter, J. Zelenka, G. Pirker and A. Wimmer, "Development of a tool for the preliminary design of large engine prechambers using machine learning approaches," *Applied Thermal Engineering,* vol. 191, no. 116774, 2021.

[17] P. Garg, E. Silvas and F. P. T. Willems, " Potential of machine learning methods for robust performance," *IFAC-PapersOnLine,* vol. 54, no. 10, pp. 189-195, 2021.