SEINÄJOEN AMMATTIKORKEAKOULU
SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

Sami Ketola

Information model for combustion engines

Thesis
Fall 2023
Automation Engineering (MSc)

SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

Thesis abstract

Degree Programme: Master of Engineering, Automation Engineering

Author: Sami Ketola

Title of thesis: Information model for combustion engines

Supervisor: Ismo Tupamäki

Year: 2023 Number of pages: 61 Number of appendices: 1

The pace of production is accelerating in maritime industry. At the same time the size of ships is getting bigger and on-board systems are getting more complex. These things present challenges during commissioning as all systems must be tested and they must work as a whole. From the automation systems' point of view there are no commonly used standardized ways for different systems to exchange data between each other.

OPC UA is widely seen as a promising solution for the communication between different systems and devices. It is a platform independent interoperability standard for secure and reliable data exchange. The lack of standardized communication methods applies to the whole maritime industry but in the thesis focus was on the communication between a reciprocating combustion engine and external systems. More specifically, the goal of the thesis was to create an OPC UA information model for Wärtsilä combustion engines.

Design science was used as the research paradigm for the thesis since it is suitable for the development of a concrete solution. The basic theories and principles behind OPC UA were reviewed in the beginning of the thesis project. In addition to the OPC UA, ISO 19848 standard for on-board data exchange was researched.

As the result of the thesis there was the first version of the information model for reciprocating internal combustion engines. ISO 19848 related information was added to the model to enable the universal identification of engine signals without any manufacturer specific knowledge. The results stated that the solution had several advantages compared to the current solution.

[1] Keywords: automation systems, combustion engines, data exchange, OPC UA, shipbuilding industry

SEINÄJOEN AMMATTIKORKEAKOULU

Opinnäytetyön tiivistelmä

Tutkinto-ohjelma: Insinööri (ylempi AMK), Automaatiotekniikka

Tekijä: Sami Ketola

Työn nimi: Informaatiomalli polttomoottoreille

Ohjaaja: Ismo Tupamäki

Vuosi: 2023 Sivumäärä: 61 Liitteiden lukumäärä: 1

Meriteollisuuden alalla tuotantotahti on kiihtyvä. Samaan aikaan laivojen koko suurentuu ja laivojen järjestelmät muuttuvat yhä monimutkaisemmiksi. Nämä asiat aiheuttavat ongelmia järjestelmien käyttöönottovaiheessa sillä kaikki järjestelmät täytyy testata ja niiden yhteentoimivuus täytyy taata. Automaatiojärjestelmien näkökulmasta katsottuna, standardoitua tapaa tietojen siirtämiseen eri järjestelmien välillä ei ole yleisesti käytössä.

OPC UA nähdään laajasti lupaavana ratkaisuna tiedonsiirtoon erilaisten järjestelmien välillä. Se on alustariippumaton yhteentoimivuusstandardi, joka mahdollistaa turvallisen ja luotettavan kommunikaation. Standardoidun tiedonsiirtomenetelmän puute koskee koko meriteollisuuden alaa, mutta tässä opinnäytetyössä keskityttiin kommunikaatioon mäntätoimisen polttomoottorin sekä ulkoisen järjestelmän välillä. Tarkemmin ottaen, työn tarkoituksena oli luoda OPC UA informaatiomalli Wärtsilän polttomoottoreille.

Tutkimusotteena käytettiin suunnittelutiedettä sillä se soveltuu käytettäväksi silloin kun ollaan luomassa jokin konkreettinen ratkaisu. Työn alussa tutustuttiin OPC UA:n perusteisiin ja taustoihin. OPC UA:n lisäksi työssä tutkittiin ISO 19848 standardia, joka on luotu laivalla tapahtuvan tiedonsiirron standardointia varten.

Työn tuloksena saatiin informaatiomallin ensimmäinen versio mäntätoimisille polttomoottoreille. Malliin yhdistettiin ISO 19848 standardin mukainen informaatio, jonka kautta moottorin eri signaalit on mahdollista tunnistaa maailmanlaajuisesti ilman valmistajakohtaista tuntemusta. Lopputuloksissa osoitettiin ratkaisun tuovan useita etuja verrattuna nykyiseen.

[1] Asiasanat: automaatiojärjestelmät, polttomoottorit, tiedonsiirto, OPC UA, telakkateollisuus

TABLE OF CONTENTS

Pictures, Figures and Tables

Terms and Abbreviations

(OPC) A&E	Alarms & Events.
(OPC) DA	Data Access.
(OPC) HDA	Historical Data Access.
COM	Component Object Model.
DCOM	Distributed COM.
DCS	Distributed Control System.
DNV	Det Norske Veritas – Classification society.
DS	Design Science Paradigm.
DSRM	Design Science Research Methodology.
ERP	Enterprise Resource Planning.
Gmod	Generic Product Model.
HMI	Human Machine Interface.
IAS	Integrated Automation System.
IIoT	Industrial Internet of Things.
ISO	International Organization for Standardization.
JSMEA	Japan Ship Machinery and Equipment Association.
JSON	JavaScript Object Notation.
M2M	Machine to Machine Communication.
MES	Manufacturing Execution System.
OPC UA	Open Platform Communications, Unified Architecture.

PLC	Programmable Logic Controller.
SCADA	Supervisory Control and Data Acquisition.
SDK	Software Development Kit.
UNIC	Wärtsilä Unified Controls, engine control system.
URI	Uniform Resource Identifier.
URL	Uniform Resource Locator – i.e., a hyperlink.
VIS	Vessel Information Model.
WSL	Windows subsystem for Linux.
XML	Extensible Markup Language.

1 INTRODUCTION

The pace of production is accelerating in the maritime industry (Wago, 2020). In addition to that, the size of ships is getting bigger and on-board systems are getting more complex. Ships consist of several systems and subsystems which all contain varying amounts of automation. All those systems will be tested within the commissioning of the ship. According to Südekum (2020), a modern cruise ship can have around 120 000 signals to be tested within commissioning and the trend is rising, so the amount of work is enormous. These challenges have been resolved by using a standardized modular approach for mechanical components, but not so much in the field of electrical engineering and software engineering (Südekum, 2018). Today different manufacturers of the on-board systems provide their own way to handle data and the integrated monitoring and control system of a ship must handle this. With the standardized way of the information exchange, it is possible to solve these challenges.

OPC UA is a platform independent interoperability standard for secure and reliable data exchange, and it is widely used within the sector of manufacturing industry. The goal of the thesis is to produce useful information about the use of OPC UA for standardized information exchange between the UNIC engine control system and external systems. UNIC is an embedded control system used with Wärtsilä 4-stroke engines. It is a distributed, modular control system consisting of multiple hardware modules which are mounted on the engine. The external systems can be, for example, the integrated automation system (IAS), propulsion control system or exhaust treatment system of a ship. Currently, a lot of hard-wired I/O connections are used between UNIC and the external systems to exchange information. In addition to that, Modbus TCP/IP or RTU can be used but this is still not a standardized solution. The intention of the thesis is to produce information mainly for Wärtsilä A&C, but also other departments within Wärtsilä may get useful information.

The main purpose is to create an OPC UA information model for Wärtsilä 4-stroke engines. The information model should describe the structure of a combustion engine and its components and to be compatible with signal naming rules and rules for metadata of signals introduced in ISO 19848 standard for shipboard data exchange as well. Although, some simplified OPC UA server applications will be developed for testing purposes, the focus is on the development of the information model.

1.1 Structure of the thesis

Further on this chapter, the target company of the thesis will be introduced. The chapters 2, 3 and 4 are dealing with the theories behind the subjects of the thesis as well as the theories and principles behind the research paradigm which was chosen. The basic principles of OPC UA as well as the history behind it is introduced in chapter 2. The ISO 19848 standard for shipboard data exchange and sensor naming rules are introduced within chapter 3. Chapter 4 represents the details and ideas of design science. The actual practical implementation part of the thesis is presented in chapter 5. The conclusion as well as the results of the thesis are introduced in chapter 6. Finally, the final words are told in chapter 7.

When writing about OPC UA terminology and definitions, italics and CamelCase writing style are used to help the reader. For the same reason, graphical notation rules provided by the OPC foundation are used when OPC UA information model examples are described. Those rules are defined in chapter 2.3.2.

1.2 Wärtsilä

Wärtsilä enables innovative technologies and solutions in the marine and energy sector globally (Wärtsilä, 2023). To enable environment friendly and successful business for customers, Wartsila’s emphasis is on sustainable technology and services. Wärtsilä employs 17 500 professionals in more than 240 locations in 79 countries. Total net sales in 2022 were 5,8 billion euros. Wärtsilä has 4 business areas which are described below.

Wärtsilä energy covers decarbonization services, future-fuel enabled balancing power plants, hybrid solutions, energy storage and optimization technologies (Wärtsilä, 2023). Wärtsilä has installed 76 GW of power plant capacity and 110 energy storage installations all over the world.

In the marine technology sector Wärtsilä is the world leader. Marine power has broad portfolio of engines, propulsion systems, hybrid technology and integrated powertrain systems (Wärtsilä, 2023). Marine power’s voyage solutions include bridge infrastructure, cloud data services, decision support systems and smart port solutions. Marine systems include gas value chain, exhaust treatment, shaft line, underwater repair, and electrical integrations.

Wärtsilä portfolio business includes units, which have limited synergies with other Wärtsilä business areas. Currently, Automation, Navigation & Control Systems (ANCS), Marine Electrical Systems and Water & Waste are included to Wärtsilä portfolio business (Wärtsilä, 2023).

2 OPC

With the OPC interoperability standard it is possible to implement secure and reliable data exchange between various systems and devices in the industrial automation space and in other industries (OPC foundation, n.d.). OPC is platform independent and thus it can be used with devices from most manufacturers.

2.1 History and background of the OPC

Since early nineties, the use of software-based automation systems has rapidly increased in industrial automation (Mahnke et al., 2009, p. 1). In the past years, development of standardized automation software has been challenging due to the huge number of different protocols, interfaces and bus systems used for accessing of a data from devices. In the old DOS days, there were same kind of challenges with printer driver applications. Then it was necessary for application developers to write a particular printer driver application for every supported printer. The problem was solved by Windows which integrated the printer support into the operating system. With this solution, the responsibility of writing these printer driver applications was moved to printer manufacturers instead of application developers.

First OPC specification, OPC DA was released in 1996. It was a standardized solution for a data exchange on HMI and SCADA systems. Development of the OPC DA was started because developers of HMI and SCADA software's were experiencing same kind of challenges that was experienced earlier with the printer driver applications (Mahnke et al., 2009, p. 1). Over time OPC Foundation, who is responsible for maintaining this standard, has published several software interfaces to perform the function of information flow from the process level to the management level (Mahnke et al., 2009, p. 3-4). There are different kind of requirements for the specifications within industrial applications and considering these, three major OPC specifications have been developed. These classic OPC specifications, which are based on the COM and DCOM technology of Microsoft, are OPC DA, A&E and HDA. The classic specifications are further introduced in the following text.

The most important OPC interface is OPC DA, and it is used with 99% of the implementations using OPC technology today (Mahnke et al., 2009, p. 4). OPC DA makes it possible to write, read, and monitor current process data via the variable, containing this data. OPC DA, like the other classic OPC interfaces, uses client-server approach for the data exchange.

OPC DA is mostly used for moving real-time data from DCSs, PLCs, and other control devices to HMIs.

OPC A&E makes it possible to receive alarm notifications and event notifications. If some process value is exceeding the limit level in the process, the client can be notified by the alarm notification (Mahnke et al., 2009, p. 5). As well, the client can be notified by the event notification if some event occurs, as in the example if an operator has changed some set point. The client can subscribe for notifications from the server. When subscribed, the client receives all the notifications triggered in the server. If needed, the number of notifications can be reduced by certain filter criteria.

As with OPC DA continually changing real-time data can be accessed, provides OPC HDA access to the data already stored. The scale is from complex SCADA system to a simple serial data logger (Mahnke et al., 2009, p. 6). The first of the three different ways for reading historical data is reading raw data from the archive, where the client has defined one or more variables and the time domain wanted to be read. When the client is reading the historical data, the return from the server is all values, up to the maximum number defined by the client. These are then archived for a specified time range. The second way is reading the values of variables with specified time stamps. The last of the three mechanisms computes aggregate values from data in the history database with a specific time domain for variables.

Classic OPC standards were successfully used with many applications, even in many such areas it was not designed for, but there were still many areas where manufacturers wanted to use standards like OPC (Mahnke et al., 2009, p. 8). However, it was not possible to use classic OPC in all cases because of the COM dependency or limitation for remote access with DCOM. Because COM and DCOM could only be used with Windows operating system, OPC UA was implemented. The idea was to develop a replacement for all existing COM-based specifications without losing any performance or features, in addition it also had to be platform independent with extensible modelling capabilities. A more accurate list for requirements can be found in Table1.

Table 1. OPC UA requirements (Mahnke et al., 2009, p. 9).

Communication between distributed systems	Modelling data
Reliability by:	Common model for all OPC data
-Robustness and fault tolerance -Redundancy	Object-oriented Extensible type system
Platform-independence	Meta information
Scalability	Complex data and methods
High performance	Scalability from simple to complex models
Internet and firewalls	Abstract base model
Security and access control	Base for other standard data models
Interoperability	

2.2 OPC UA today

Today, OPC UA can be used with every component in industry processes. Those components can be sensors, actuators, control systems, MES, and ERP-systems including IIoT and M2M as well as Industry 4.0 and China 2025 (OPC foundation, 2017, p. 8). Those systems are intended to be used for information exchange and to control processes in industrial domain. OPC UA specifies common infrastructure for information exchange with following features:

1. Information model that represents structure, behavior, and semantics.
2. Message model to be used between applications.
3. Communication model that exchanges data between endpoints.
4. Model to fill the requirements set for the interoperability of the systems.

2.3 Basics of OPC UA

Just to mention at first, according to Industry40tv (2020), OPC UA should not been compared to industry communication protocols since it is not a protocol at all. OPC UA was created to solve entirely different problems than industrial communication protocols.

OPC UA consists of two different layers which are an information model layer and a communication layer (Unified Automation, 2022a). Different building blocks of OPC UA belong to one of these. The information model layer exists for application specific information that is modeled within the address space. To enable the capability of communication for OPC UA applications, the communication layer is implemented by the SDK. The SDK and other software layers are introduced in chapter 2.3.1. These and other basic building blocks can be seen in the figure 1.

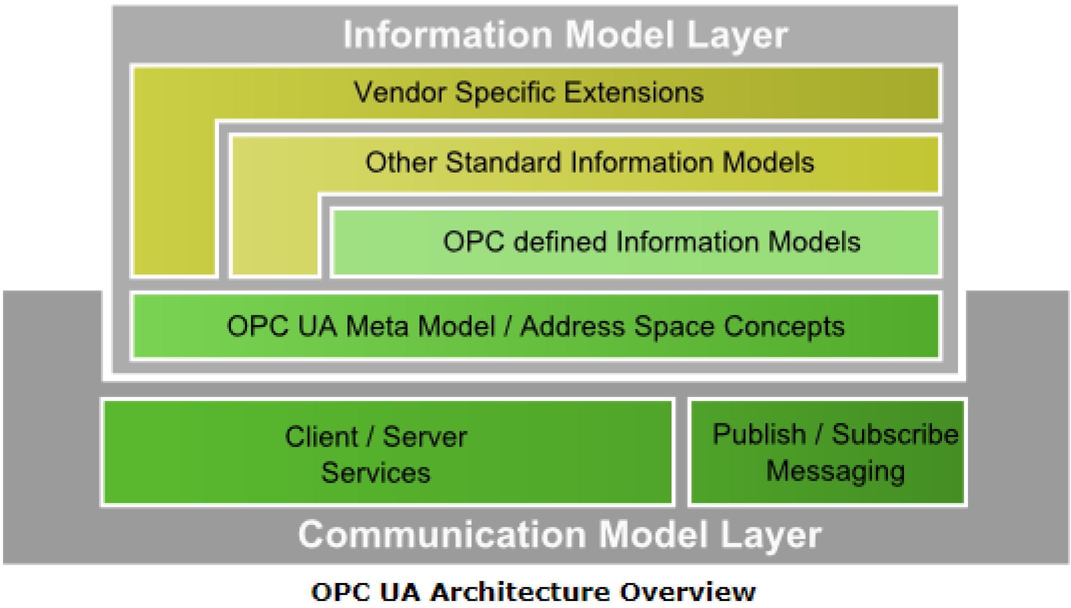


Figure 1. OPC UA basic building blocks (Unified Automation, 2022a).

The OPC UA meta model defines the basic concepts of the information model layer. In addition to that It define rules for representing of the object-oriented address space and thus the meta model is the foundation for the information models. More about the address space and information models will continue in chapter 2.3.2.

OPC UA services for the client / server defines the information exchange between UA applications. Communication between the client and the server is implemented by a requestresponse pattern. For example, the client requests the list of variables to read from the server and then server responses with the values for the variables. In addition, to enable information exchange, another job of the services is to manage and optimize the communication.

Messaging model for the publish / subscribe is alternative concept for the client / server implementation. It is a one-to-many communication model, where a publisher is publishing data or events to the network, and it does not matter how many subscribers there are. So, it is completely different compared to the client / server implementation which is a one-to-one communication. Network switches and message brokers are responsible for distribution of the messages in the publish / subscribe concept. The publish / subscribe approach is better if many receivers are interested about the same information. Real-time communication is also possible with the publish / subscribe model. The client / server approach is better if wide information models are needed.

In addition to these building blocks, OPC UA also includes a built-in security model that includes an application authentication, message signing and message encryption on the protocol level. User authentication is defined at the service level and user authorization and permission handling is defined at the information model level.

2.3.1 OPC UA software layers

Typically, three software layers are used with OPC UA applications (Unified Automation, 2022c). These layers, the application-layer, the SDK-layer, and the stack-layer are shown in Figure 2.

A system that wants to consume or provide data via OPC UA is an OPC UA application. With the client / server concept, an application that exposes some data from its own information is called a UA server. When the situation is the opposite, where an application is consuming data from another application, it is called a UA client. Applications can also consist of both, the UA server, and the UA client.

The purpose of the client or server SDK is to enable common OPC UA functionality as a part of the application layer. It also facilitates faster interoperability for an application and thus reduces the development effort.

OPC UA stack is there to define the details of communication and to handle communication protocols (OPC foundation, 2022d, p. 5). The stack is further divided in three different layers, which are the data encoding layer, the security protocol layer, and the transport protocol layer.

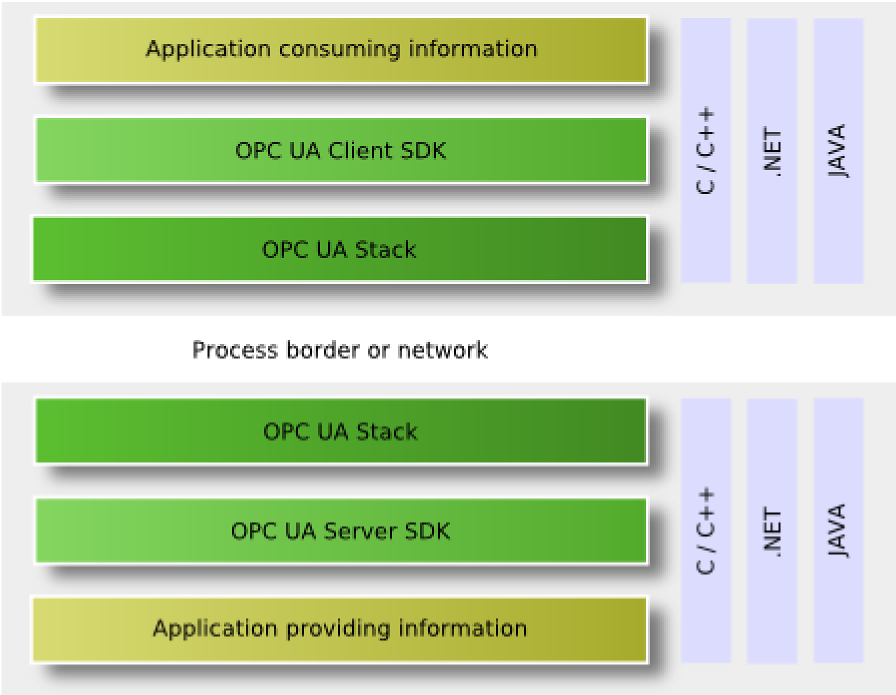


Figure 2. Software layers (Unified Automation, 2022c).

2.3.2 OPC UA address space model

According to Pipero (2018) the idea of the OPC UA address space model is the unified object model. The reason behind the object-based model is that complexity in the industry automation has increased over the years. The object-based model is capable to handle this complexity and it also makes possible to describe information in different ways at the same time. Objects are instances of physical world aspects, and they can contain variables which holds data. Variables maps to the classic OPC DA and HDA, where variables with the current or historical data can be read and written. Objects can as well hold methods, and via methods, functions can be invoked. In addition of those, there is possibility for event notifications and that maps to the classic OPC A&E. As an example, event notification can be triggered if an operator has changed some setpoint. To describe relationship between different objects, references are used. Example of the OPC UA object is represented in figure 3.

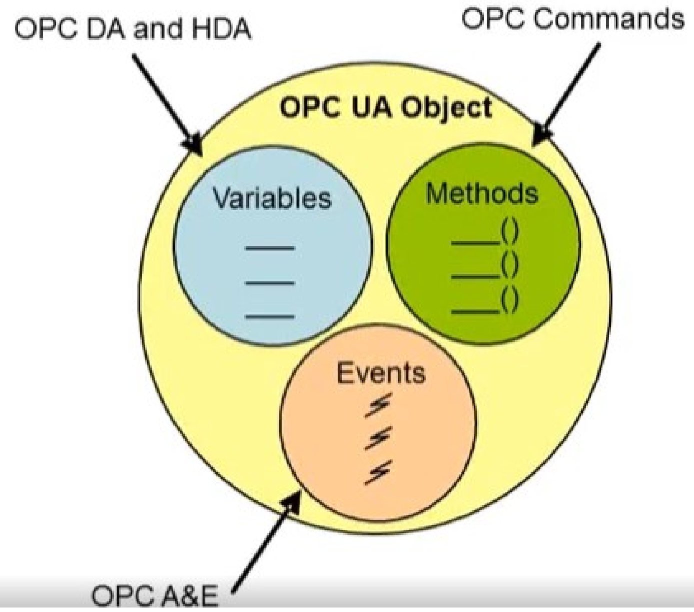


Figure 3. Object model (Damm, 2014).

The way how the data is presented in the address space is implemented via nodes. So, everything in the address space is a node (Damm, 2014). OPC UA defines a non-extensible list of 8 node classes which are: *Object*, *Variable*, *Method,* and *View* nodes. The rest of them are type related nodes and they are: *ObjectTypes*, *VariableTypes*, *DataTypes* and *ReferenceTypes*.

Referring to Pipero’s thoughts (2018), the idea is to have a model that in addition to just transfer the information, it also has a method to describe what the information is supposed to represent. The method enables identification and qualification of the node. To enable these features, each node in the address space has set of so-called attributes and to describe dependencies to other nodes, they are connected by references (OPC foundation, 2022e, p. 5). Attributes and references are described in the figure 4. Every node class have set of attributes and some attributes are common for all node classes. These are: *NodeId*, which is different for every node, *NodeClass*, *BrowseName*, *DisplayName,* and *Description*. The Idea of these common attributes is to identify node and make it possible to find nodes by UA clients. In addition to those, there are attributes, which are the node class dependent. For example, the variable node has an attribute for *Value* and *DataType* to better describe the value of the data inside the variable.

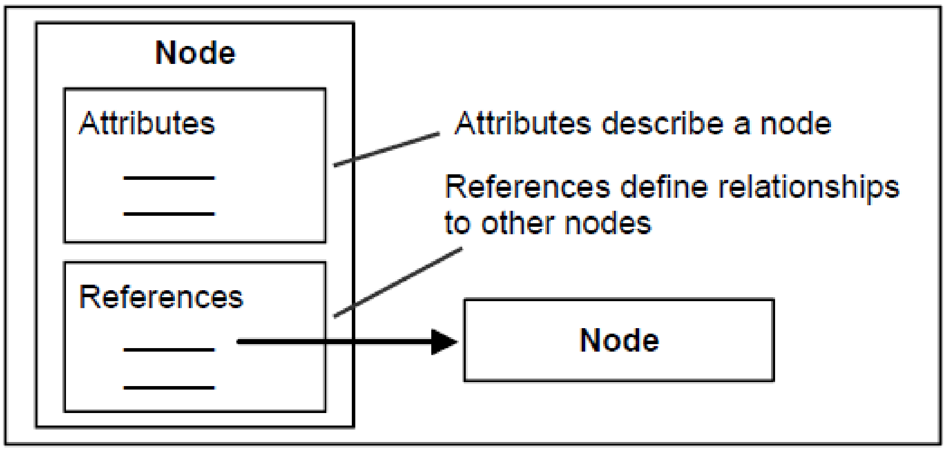


Figure 4. Attributes and references (OPC foundation, 2022e, p. 6).

In the figure 5, some ideas of the address space model can be seen. It can be thought of that the information of the address space is located inside of two different places, in the type-space and the instance-space. With the help of types in the type-space, information in the address space can be found and identified, so types are part of the meta model (Pipero, 2018). Information itself, like the value of some variable is available in the instance space. In the other words, types are kind of templates for instances of them located in the instance space.

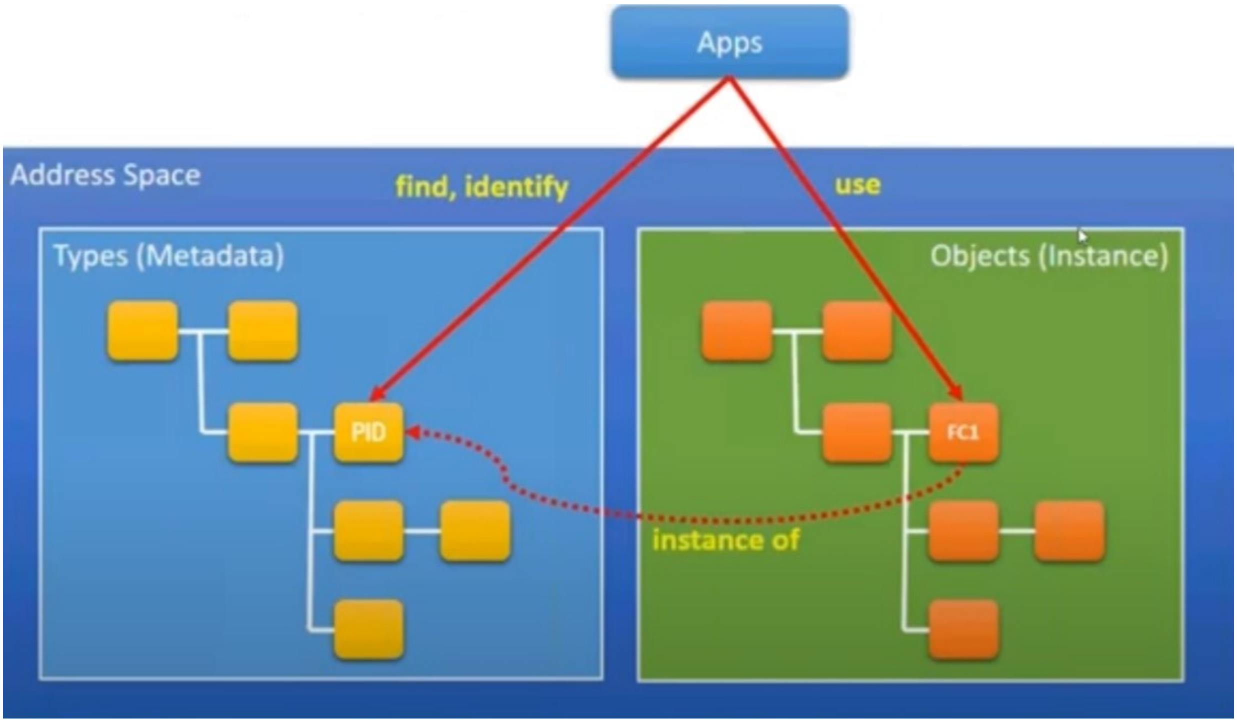


Figure 5. Types and instances (Pipero, 2018).

To illustrate structure of different information models, OPC foundation (2022e) provides graphical notation rules for node classes in UA Specification part 3, which can be seen also in the tables 2. UA Specification part 3 also defines graphical notation rules for the reference types, which can be seen in the table 3. Those rules are used within the thesis when examples are described. OPC Foundation provides set of shapes for Microsoft Visio, and they were used for graphical representation of the information model developed during the thesis.

Table 2. Graphical notation 1 (OPC foundation, 2022e, p.104).

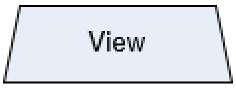
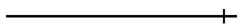
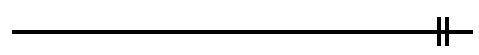
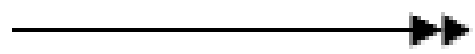
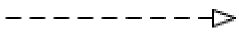
NodeClass	Graphical representation	Comment
Object		Rectangle including text representing the string-part of the DisplayName of the Object. The font shall not be set to italic.
ObjectType		Shadowed rectangle including text representing the string-part of the DisplayName of the ObjectType. The font shall be set in italic.
Variable		Rectangle with rounded corners including text representing the string-part of the DisplayName of the Variable. The font shall not be set in italic.
VariableType		Shadowed rectangle with rounded corners including text representing the string-part of the DisplayName of the VariableType. The font shall be set in italic.
DataType		Shadowed hexagon including text representing the string-part of the DisplayName of the DataType. The font shall be set in italic.
Reference-Type		Shadowed six-sided polygon including text representing the string-part of the DisplayName of the ReferenceType. The font shall be set in italic.
Method		Oval including text representing the string-part of the DisplayName of the Method. The font shall not be set to italic.
View		Trapezium including text representing the string-part of the DisplayName of the View. The font shall not be set to italic.

Table 3. Graphical notation 2 (OPC foundation, 2022e, p.105).

Reference-Type	**Graphical representa-tion**	**Comment**
Any symmetric Reference-Type		Symmetric ReferenceTypes are represented as lines between Nodes with closed and filled arrows on both sides pointing to the connected Nodes. Near the line has to be a text containing the string-part of the BrowseName of the ReferenceType.
Any asymmet-ric Reference-Type		Asymmetric ReferenceTypes are represented as lines between Nodes with a closed and filled arrow on the side pointing to the TargetNode. Near the line must be a text containing the string-part of the BrowseName of the Refer-enceType.
Any hierar-chical Reference-Type		Asymmetric ReferenceTypes that are subtypes of HierarchicalReferences should be exposed the same way as asymmetric ReferenceTypes except that an open arrow is used.
HasCompo-nent		The notation provides a shortcut for HasComponent References shown on the left. The single hashed line must be near the TargetNode.
HasProperty		The notation provides a shortcut for HasProperty References shown on the left. The double hashed lines must be near the TargetNode.
Has-TypeDefinition		The notation provides a shortcut for HasTypeDefinition References shown on the left. The double closed and filled arrows have to point to the TargetNode.
HasSubtype		The notation provides a shortcut for HasSubtype References shown on the left. The double closed arrows must point to the SourceNode.
HasEven-tSource		The notation provides a shortcut for HasEventSource References shown on the left. The closed arrow must point to the TargetNode.
HasInterface		The notation provides a shortcut for the HasInterface References shown on the left. The closed arrow shall point to the TargetNode.

Considering the notation rules and the background of OPC UA, more detailed example is described in the figure 6. The upper part represents *ObjectType* which is a template for the instance of it (OPC foundation, 2022a, p. 18). In the example, *PersonType ObjectType* has two children, *LastName* and *FirstName*. All instances of *PersonType* shall have these children also. OPC UA have possibility for subtyping. It means that an existing type can be used as a base and extended with additional features. In the example, *PersonType* is a subtype of *BaseObjectType* and is extended with *FirstName* and *LastName* variables. The instance of *PersonType* is further extended with one more variable and all variables have values with *DataType* of a string.

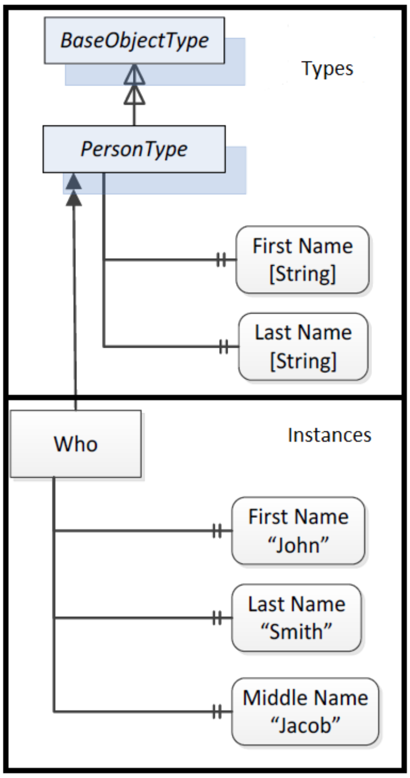


Figure 6. Address space example (OPC Foundation, 2022a).

In the beginning of the chapter 2.3 can be found the figure 1. From the figure can be seen that information models consist of three different information model layers. The nethermost is OPC UA built in information model, and it provides the structure for all information models using OPC UA (Unified Automation, 2022b). It provides following characteristics:

- it provides entry points into the address space for clients to navigate through types and instances of an OPC UA server.
- base types to be the root for different hierarchies of types.
- extensible types as built in.
- the server object for enabling diagnostic information and capability.
- access to a data which maps to the classic OPC DA mentioned earlier.
- state machines.
- alarms and conditions which maps to OPC A&E mentioned earlier.
- programs.
- security configuration and management of OPC UA.
- capability to handle file transfer.
- identification, management, and configuration of a device.

The midmost of the layers in the figure 1 is for companion information models. Companion information models can be domain, use case, machine or device specific models made by some organization in cooperation with experts from the OPC Foundation. Big part of these standard models for a specific scope are mappings of existing standards to OPC UA. Examples of these are: PLCopen, ISA95, BACnet or a device descriptions like Field Device Integration.

The top one of these layers is for manufacturer specific information models. Product manufacturers can create their own information models although it is recommended to use standard information models as far as possible. Even if manufacturer specific models are used, one or more standard information models can be used as a base.

3 SHIPBOARD DATA STANDARDIZATION

The data produced by the ships' sensors plays an increasingly large part in the operations of ships (Låg et al., 2021, p. 1). The data produced by different shipboard systems is primarily used on-board but can be sent to a shore for improved ship-to-shore connectivity as well. Huge investments for digitalization by all maritime players have been made over the last years. Unfortunately, it has been showed by pilot projects that there are still obstacles against efficient use and sharing of a sensor data in new applications. One of the major obstacles has been the lack of standardization.

According to International Organization for Standardization (ISO, 2018, Introduction) there have been a standard for data exchange of a navigational equipment but not for other on-board components and systems. Exchanging of a non-standardized data between applications requires name-based aggregation and format mapping and this requires a huge amount of manpower.

To solve these problems, ISO have published two standards which are ISO 19847 and 19848 (Låg et al., 2021, p. 2). ISO 19847 is the standard for the shipboard data server, and it defines the requirements for the functionality and performance of data handling. In the figure 7 can be seen that ISO 19847 specifies standard input and output protocols and provides a "neutral ground" for the data storage between data consumers and providers.

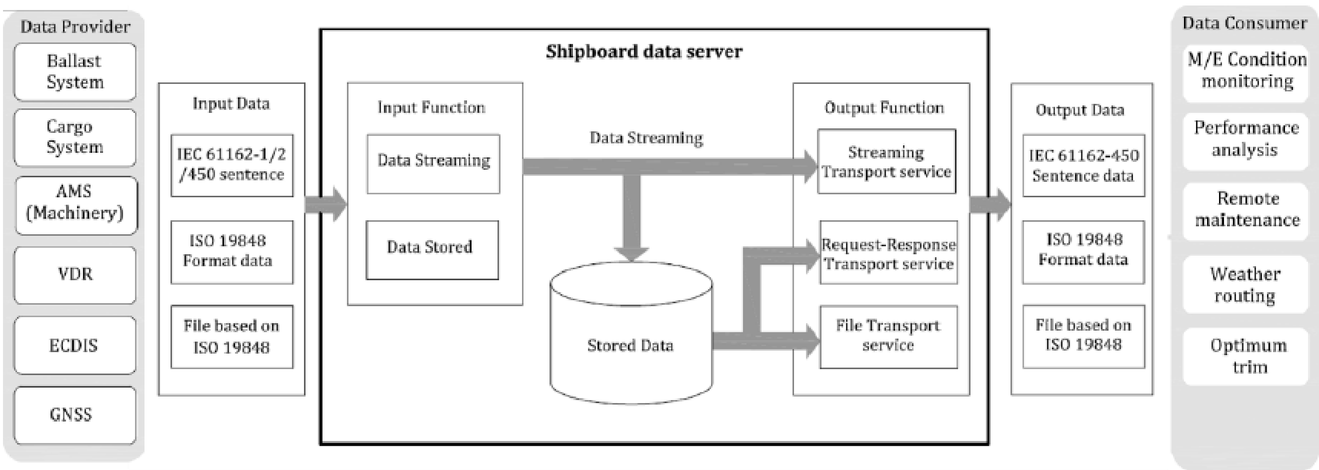


Figure 7. Shipboard data server model (Låg et al.,2021, p. 2).

In the figure 7 can be seen also that the ISO 19848 standard specifies the data concepts and structures which the data server defined in ISO 19847 must use. ISO 19848 defines three areas and the first of them is the structure of a time series data which is the actual reading or measurement data in the form of tabular or event data. If a high-frequency sampling interval of several channels for physical measurements are needed, the tabular data structure is better. A suitable data structure for a low-frequency asynchronous data source is an event data. The second area that ISO 19848 defines is the standard structure for a meta data of a sensor and it is called the Data Channel Property (Låg et al., 2021, p. 3). According to International Organization for Standardization (ISO, 2018, Chapter 5.3) in the table 4 can be seen what is defined by the Data Channel Property.

Table 4. Data Channel Property (Låg et al., 2021, p. 3).

Data Channel Type	Type	*E.g Instant, Max, Min, setpoint*
	Update Cycle	*E.g 1 sec*
	Calculation	*E.g 60 sec*
Format	Type	*E.g Decimal, Integer, Boolean, String*
	Restriction	*E.g Enum, FractionDig, Length, MaxLength*
Range	Low	*E.g 0*
	High	*E.g 120*
Unit	Unit Symbol	*E.g kW, kPa, s*
	Quantity Name	*E.g Active Power Pressure*
Quality Cod-	*E.g OPC-Quality, IEC 61125-STATUS*	
Name	*E.g Main Engine Revolution*	
Remarks	*E.G Location: ECR Manufacturer Wärtsilä*	

The last area what ISO 19848 defines is the standard method for construction of so called UniversalID (Låg et al., 2021, p. 3). The UniversalID is a URI and could be a unique URL and defines the universal identification for a sensor or signal connected to a device or component. It is possible to use two different naming rules with the UniversalID which are: JSMEA or DNV. The UniversalID on top in the figure 8 is implemented with the JSMEA and the bottom with the DNV rules.

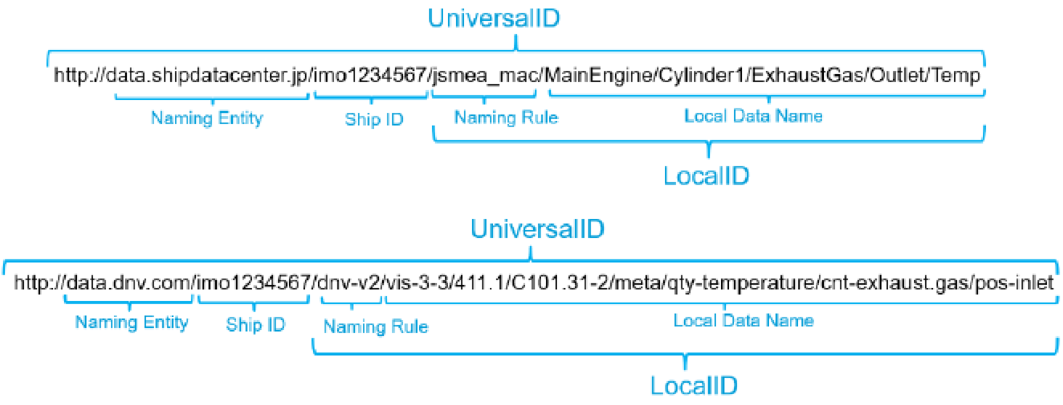


Figure 8. UniversalID (Låg et al., 2021, p. 3).

The DNV's naming rule is based on DNV's VIS model which has been used as a basis for classification processes (Låg et al., 2021, p. 4). An important part of the VIS is a functionally oriented Gmod model. Gmod is a hierarchical model that describes functional breakdown of a vessel. It also provides a library of product types that may be installed on the vessel. All items in the model have unique names and codes. With the help of the Gmod, sensors have a unique reference to the component they are connected to. The reference can be used in a sensor-related data such as the measurements, meta data, configuration, analysis, or results from processing the data. An example of the structure of the Gmod model can be seen in the figure 9 where green ones are functional items and red ones are physical items.

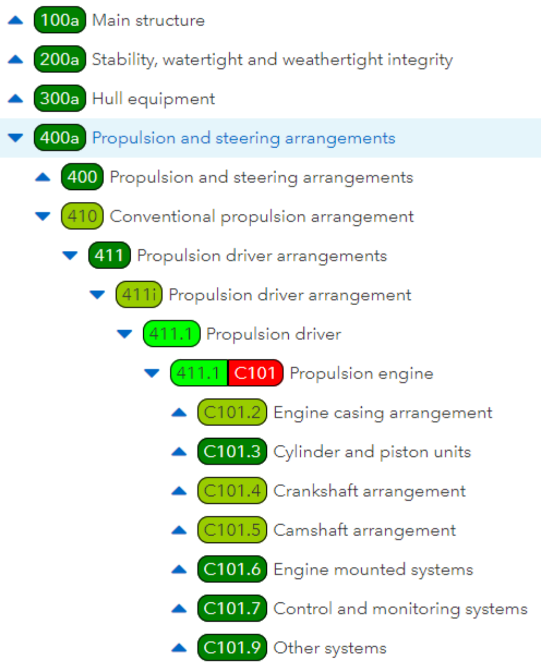


Figure 9. Gmod model (DNV, 2023).

Since the Gmod model only describes functions and products shipboard, it is necessary to have some way of describing the sensor itself (Låg et al., 2021, p. 8). As well there can be several sensors attached to the same function or product. For that reason, DNV has defined eight meta tags for describing a sensor. These meta tags, which are described in the following list, are optional but at least one of them shall be used.

- "qty" (Quantity) Describes what is the quantity that is measured, examples: Temperature, pressure, and humidity.
- "cnt" (Content) Describes what is substance that is measured, examples: Exhaust gas and lubrication oil.
- "calc" (calculation) Describes calculations that may be applied to a measurement, examples: Maximum, mean and sum.
- "state" Can be used for statuses like for example: Closed, opened, and running.
- "cmd" (Command) Can be used for commands, for example: Stop and start.
- "type" Describes the type of data, examples: Set point and control output.
- "pos" (Position) Describes a relative position of the sensor, examples: Outlet, inlet and upper.
- "detail" Is a free text for important information about the sensor that cannot be described with other elements.

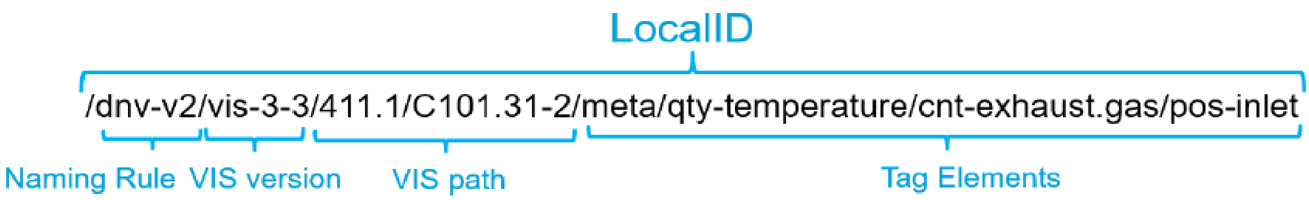


Figure 10. LocalID with meta tag (Låg et al., 2021, p. 9).

From the figure 10 can be seen detailed structure of the LocalID. The VIS path describes functions and products defined by the Gmod model and the Tag Element describes the sensor or signal itself and its meta information.

4 INTRODUCTION OF THE RESEARCH PARADIGM

According to Jokinen (2021) the design science is applicable if it is intended to produce a workable solution for known, practical problem. The solution can be for example the actual device, software, operating process, or some other construction. Järvinen (2006) is also of the opinion that the design science is applicable for practical problems, but on the condition that the implemented construction must be well justified, and the quality of the construction must be verified. The aim of the thesis was to produce a standardized OPC UA information model for internal combustion engines since there is no similar solution available. Considering the aforementioned information, the design science research paradigm was selected. The theory and principles behind the design science paradigm are introduced further within the chapter.

The field of information systems science must develop knowledge about information technology management and its use for management and organizational purposes (Hevner et al., 2004, p. 76-78). There are two separate paradigms for acquiring such knowledge: Behavioral science and design science. Behavioral science seeks to develop and justify theories that explain or predict organizational and human phenomena concerning information systems. Approach of the design science is different. It seeks to create innovations that define the ideas, technical capabilities, practices, and products through which the analysis, implementation, design, management, and use of information systems can be efficiently and effectively achieved. Although knowledge from the behavioral science is used as a base also for the design science, Peffers et al. (2007, p. 46-47) argues that traditional research paradigms borrowed from social and natural sciences are not enough for the field of information systems science.

The figure 11 represents Hevner's et al. (2004, p. 80) framework for combining of the behavioral science and the design-science. In the figure, the environment defines the business need or "problems" for which the researcher wants to find a solution. The goals, problems, tasks, and opportunities are formed through peoples, organizations, and their technologies. The knowledge base provides a theoretical basis for the research process. It provides foundational theories, instruments, models, constructs, frameworks, and instantiations used in the development and build phase. Methodologies used in the justify and evaluate phase comes also from the knowledge base. Finally, the design science part is between the environment and the knowledge base, providing building and evaluating of the innovative artifacts with an iterative and incremental process. Through these innovative artifacts, new information is created for the knowledge base.

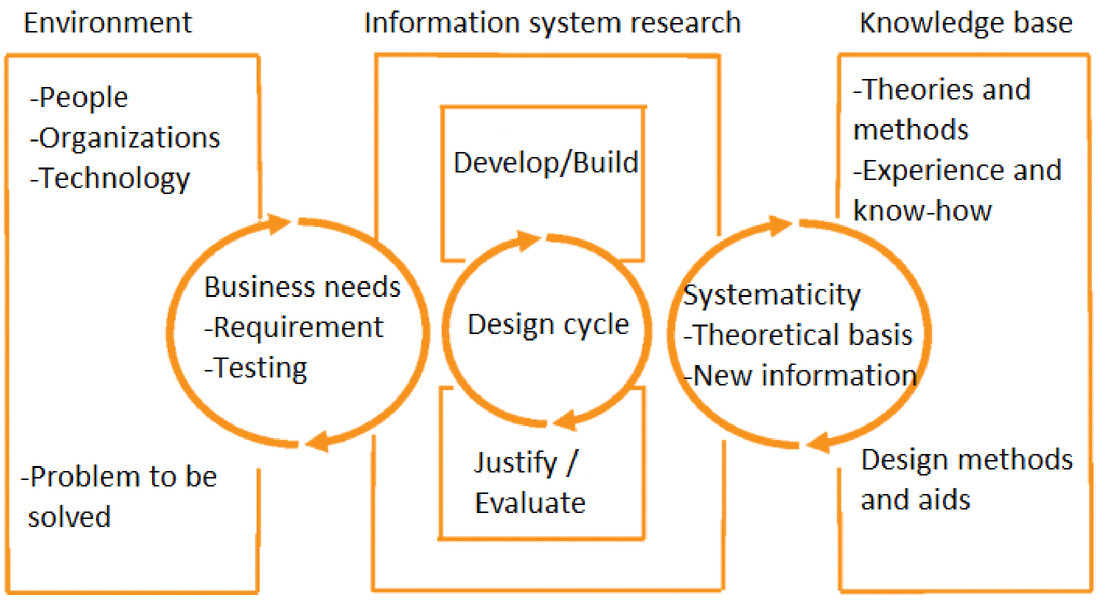


Figure 11. Design science framework (Jokinen, 2021).

Hevner et al. (2004, p. 82-90) defines seven guidelines for the effective design science process. The guidelines are:

1. **Design as an artifact**. The product developed through design science is a construction. The construction can be, for example the actual device, software, operating process, or some other construction. The construction can also be a design method, a design aid, or an abstract construction such as an operating process.
2. **Problem relevance.** The artifact must solve a real, practical problem. The idea is to develop technology-based solutions to important and relevant business problems. The design work starts with identification, delimiting and describing of the problem that is intended to be solved.
3. **Design evaluation.** Evaluation is an integral part of the research process, especially in the design of IT artifacts, which must be rigorously tested for functionality, consistency, accuracy, reliability, and usability, among other quality attributes. A description of the functional requirements must be drawn up for the construction and the acceptance criteria must be defined. In addition of meeting the functional requirements, construction also must solve the original problem which was described within guideline 2.

4. **Research contributions.** The results, developed through design science should also be evaluated from the point of view of the scientific value. Particularly in the areas of the design artifact, design construction knowledge, and design evaluation knowledge. The contributions can be based on the novelty, generality, and significance of the designed artifact.
5. **Research rigor.** Applying of the design science requires a systematic and disciplined approach. Systematicity includes both, application of rigorous methods and construction and evaluation of the design artifact. In addition of results, theories, and the artifact itself, the research paradigm, methods, and research plan must also be described as part of the research report. The specification of construction requirements must be prepared precisely, and the obtained results must be compared with the specification of requirements. So basically, the researcher must understand the whole research process in detail and finally one must evaluate and describe the added value brought by one's own design work.
6. **Design as a search process.** The design work itself is ideating and experimenting with different solutions with the help of iterative process of the design science. Within the design process one should familiarize with previously known solutions and methods and apply them in new conditions. The functionality of the developed artifact is ensured by testing and comparing results towards requirements.
7. **Communication of research.** Research process and results needs to be presented to both technology-oriented and management-oriented audiences in a clear way, so that the reader can evaluate the consistency of the work and the usability of the work results. The results should be published to enable the results to be used in other studies as well.

Because there has been lack of clear and closer to practice instructions, Peffers et al. (2007, p. 50-56) provides a process model for the design science, called Design Science Research Methodology (DSRM). The methodology focuses on purposeful seeking of a solution to a problem through six steps of the DS research. The process model aims to meet three objectives: provide a nominal process for the DS research, build upon prior literature, and provide a mental model or template for research outputs. Those six steps that also can be seen in the figure 12 are:

1. **Problem identification and motivation.** Defining the problem is important because the definition is used for development of the artifact during the process. In addition of defining the problem, the aim is to create value for the solution also from the audience's point of view. Creating the value is called motivation because the goal is to motivate researchers and the audience. The better the problem is understood, the better is understood also the value produced by the solution.
2. **Define the objectives for a solution.** The idea of this phase is to define feasible goals for the problem defined in the previous activity. Goals can be quantitative, such as a measurable improvement to the previous solution. As well, the goals can also be qualitative, in which case it can be a description of how the artifact is expected to support the solution to the problems that previously was not considered.
3. **Design and development.** The artifact itself is built in this phase, during which the contribution of the research is brought up. The construction of the artifact and the evaluation process (step 5) are at the center of the design science, and it must rely to the knowledge of a theory. The process is iterative, and the artifact can be developed and evaluated several times until the solution is good enough.
4. **Demonstration.** Within the demonstration phase, intention is to illustrate the use of the artifact and how the artifact solves the problem it was built for. To perform the demonstration, multiple methods can be used. For example, by using the artifact experimentally, simulating, as a case study or in another decent way.
5. **Evaluation.** This phase evaluates the demonstration results in relation to the defined research problem and the client's goals and in relation to the needs. The assessment depends on the nature of the artifact and what affects the artifact in the environment. Therefore, the assessment may involve quantitative or qualitative. After the evaluation, researcher can decide if to continue to the next step or going back to the design and development phase (step 3).
6. **Communication.** The problem that the artifact is supposed to solve, is important to communicate to the audience. The artifact’s usefulness, novelty, and the rigor of its design must also be stated. Communication usually happens to both the technology-oriented audience and the management team.

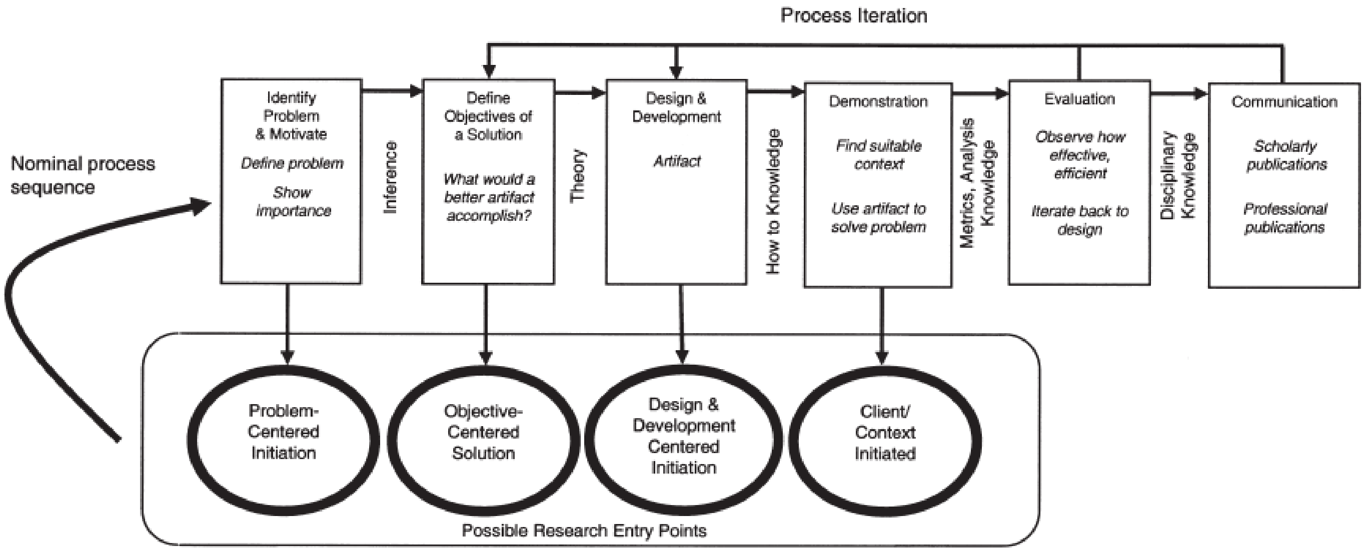


Figure 12. DSRM model (Peffers et al., 2007, p. 54).

According to Peffers et al. (2007, p. 56), although the process is nominally structured in a sequential order, it can be as well started at any step within steps 1 – 4 and move outward. It depends on what the approach is in each case.

5 IMPLEMENTATION

This chapter handles the practical part of the thesis. The problem identification and goals of the research are introduced first by comparing them to design science principles and to DSRM process. The rest of the chapter is dealing with the practical development of the artifact.

5.1 Environment and problem identification

This chapter maps the environment part of Hevner et al. (2004, p. 80) framework shown in the figure11 in the previous chapter. In addition to that it also maps the DSRM model step 1, problem identification and motivation.

As it has already been stated in chapter 1, ships are constructed from many different systems and sub-systems and produced by many different manufacturers. These systems produce data that is used shipboard and can be sent to shore also. In addition, these systems can produce data for other systems or consume data from them. These days different manufacturers of on-board systems do not usually provide standardized ways to change data with other systems.

In addition to combustion engines, Wärtsilä is the manufacturer of multiple different shipboard equipment, for example controllable pitch propeller systems and selective catalytic reduction systems. Both these systems are used in connection with the combustion engine, and they must communicate with each other. Communication between these systems within Wärtsilä can be also improved and standardized.

When talking about industry 4.0, OPC UA has been seen as one of the key enablers (Hoppe, 2017). Although, according to Mahnke et al. (2009) OPC UA has been successfully used for many years already within industry sector, it still has not been used widely within the maritime sector. OPC foundation provides many domain and device specific, so called companion specifications which can be used as a base for case specific information models, but there are not yet available specifications that are especially suitable for maritime use (OPC foundation, 2023b). However, some operators within maritime industry have applied OPC UA to their solutions and they believe that OPC UA is a potential and future proof solution (Südekum, 2018; Macgregor, 2017; VDMA, 2023; Phoenix contact, 2023).

Also, within Wärtsilä, OPC UA is seen as a future proof technology for communication between systems and Wärtsilä already provides implementations for certain purposes. OPC UA communication for the internal communication of Wärtsilä UNIC engine control system has already been investigated and the investigation is still ongoing (Ala-Hynnilä, 2022).

So, the environment where the problem and requirements come from is the whole maritime industry including shipyards and manufacturers of ship components. But within the thesis the environment is limited to Wärtsilä A&C department. As discussed above, the problem is lack of standardized communication between different on-board systems generally. However, the focus within the thesis is limited to a communication between a reciprocating internal combustion engine and external systems. For this purpose, OPC UA is being studied.

5.2 The goals of the solution

As already mentioned in the previous chapter, OPC UA is widely seen as a promising solution for a communication between different systems and devices. OPC UA covers very wide range of different features, and it is not possible to study all the features in the thesis project. Thus, the goal of the thesis is limited to the implementation of the OPC UA information model for reciprocating internal combustion engines. The solution must be applicable to different Wärtsilä engine types and cylinder configurations. It should also be possible to create an engine configuration specific information model with minimal amount of a manual work. More specific requirements for the information model are defined in the appendix 1.

Currently, Modbus TCP/IP or RTU protocols can be used for the communication between an engine and external systems. The idea of the artifact created through the design science process is to create something new or improve things compared to the previous solutions. According to Turner (2020) the advantages of OPC UA compared to Modbus are following:

- it is a platform independent and can thus be used with most devices when compared to Modbus which need a compatible platform.
- OPC UA has a built-in security, which Modbus does not contain.
- with OPC UA, clients can browse the information available in the server, so there is no necessary need to prepare the information model for a client application manually in advance.

- in addition to just reading the value of some sensor, with OPC UA it is possible to include the metadata of a sensor to the information model, such as a description, units, range and so on. With Modbus this is not possible and thus the information must be provided some other way as, for example, with a manual list of the information in advance.
- even complex information structures can be modeled in a way that is needed in certain cases.
- OPC UA can be used between many servers and clients simultaneously, making it scalable.

Nowadays quite many hard-wired I/O connections are used to exchange information between engines and external systems. There might be multiple legacy reasons why so many I/O connections are still used. However, if there is a standardized and reliable communication method available, those I/O connections can be at least reduced.

To reach those goals, in addition to following the DSRM process steps defined by Peffers et al (2007), seven guidelines for effective design science process defined by Hevner et al. (2004) will be observed. The knowledge base part of the design science process is presented in the figure 11. The knowledge base provides theoretical basis used within the research process. The significant knowledge base for the thesis is OPC Foundation as they define the rules and frames for OPC UA information models. They also provide companion information models which can be used as a base in certain cases.

5.3 Development of the artifact

As discussed earlier, the artifact which was going to be implemented within the thesis was OPC UA information model for reciprocating internal combustion engines. Although the creation of the actual OPC UA application has been decided to be left out of scope, it still needs some attention. As it turns out within chapter 2, OPC UA applications can be implemented either with a client / server approach or with a publish / subscribe approach. According to Kominek (2017) the client / server approach is good for lower frequency applications with 10 ms and above the data update interval while the publish / subscribe approach is capable of exchanging data at an update interval of 1 – 10 ms. Within Ala-Hynnilä's (2022) research, OPC UA with the publish / subscribe approach was researched because the internal communication of UNIC is time sensitive. However, the communication between engines and external systems is not as time critical, so the client / server approach is fine with that. An internal combustion engine is complex and includes many sensors and actuators with different relations to each other and thus a complex information model is needed. According to Unified Automation (2022b) only the client / server approach provides full access to all information in the address space and hereby provides possibility to represent complex information models. Based on this information the client / server approach was decided to be used for the communication between engines and external systems.

Up to this point in the thesis it has emerged that the OPC UA information model can hold useful metadata about everything the information model is representing. But the main purpose of OPC UA information model is to offer an opportunity for the representation of real-world objects. Within the thesis that object is a reciprocating internal combustion engine with all the auxiliary devices, sensors and actuators connected to it. Thus, the following chapter provides more information on this with a short introduction of a reciprocating internal combustion engine.

5.3.1 Basics of a reciprocating internal combustion engine

With the help of a crank mechanism, reciprocating internal combustion engine converts the energy of the internal combustion into a rotating motion and the resulting torque (Bauer et al., 2003, p. 409). The mechanism that converts the energy of combustion to a rotating motion, consists of a cylinder liner, a piston that moves back and forth within the liner, a connecting rod that is connected to the piston from one end and to the last part, a crank shaft from the other end. The previously mentioned basic parts of a 4-stroke diesel engine can be seen in figure 13. In the past internal combustion engines were mechanical machines, and the combustion event was controlled by mechanical parts. Today when the features required from the engines are much higher, electronic engine control systems are used. The main purpose of the engine control system is to control the combustion event. The combustion event can be controlled, for example, by controlling the duration and timing of the fuel injection or by controlling the timing and duration of intake and exhaust valve operations. In addition to controlling the combustion event, the operation of engine parts is monitored. For example, the temperatures of a cylinder liner, big end bearing, main bearing, intake air and exhaust gas are monitored.

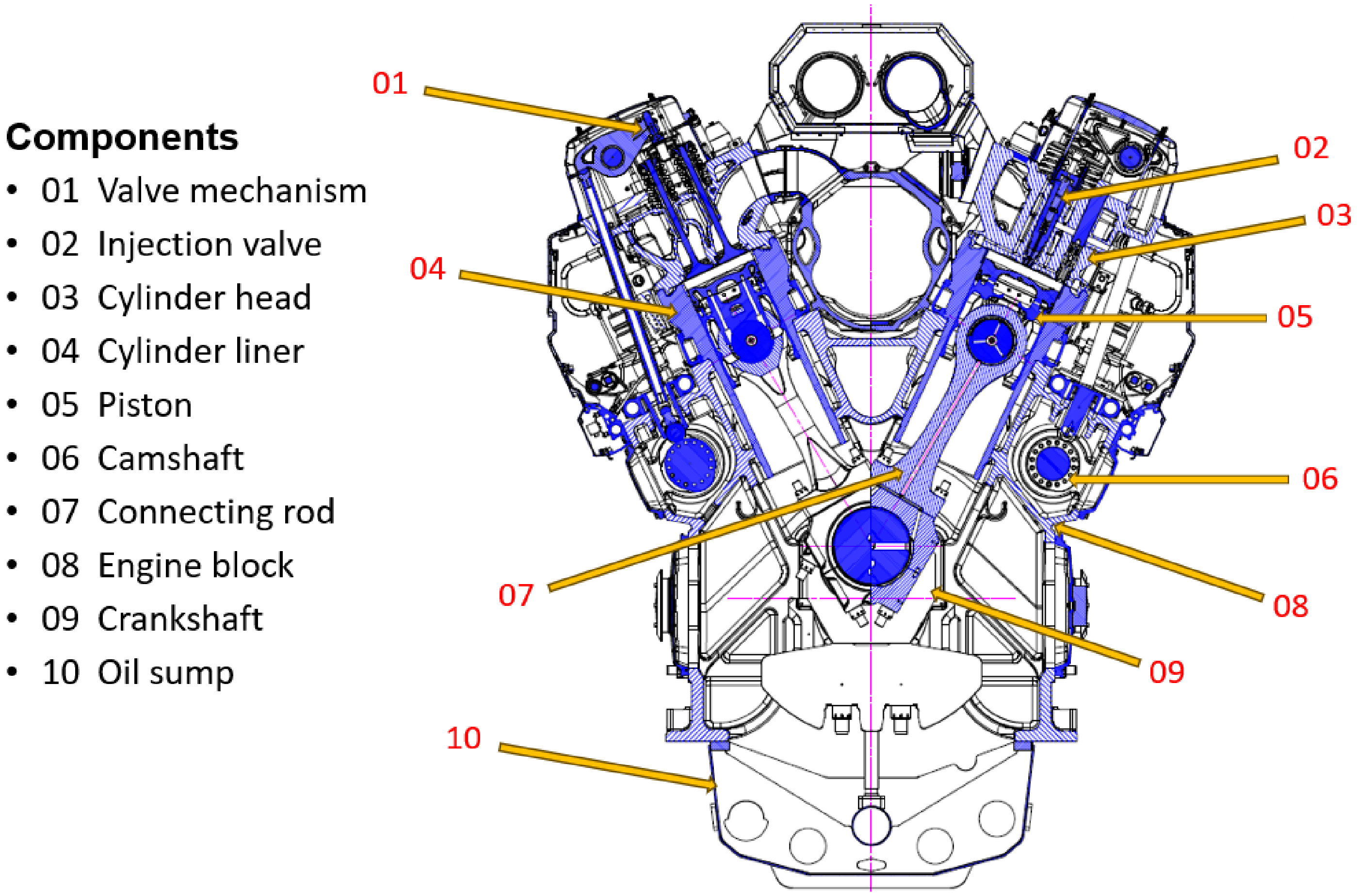


Figure 13. 4-stroke diesel engine parts.

The complete Wärtsilä engine can have up to 20 pieces of cylinder units to be controlled and monitored by the UNIC engine control system. Previously mentioned was only internal operation part of an engine. Engines also have several auxiliary components that are also controlled and monitored by the engine control system, examples of those are a charge air system including turbocharger(s), a cooling system, and a generator system. Some components of a Wärtsilä 4-stroke engine can be seen in the figure 14.

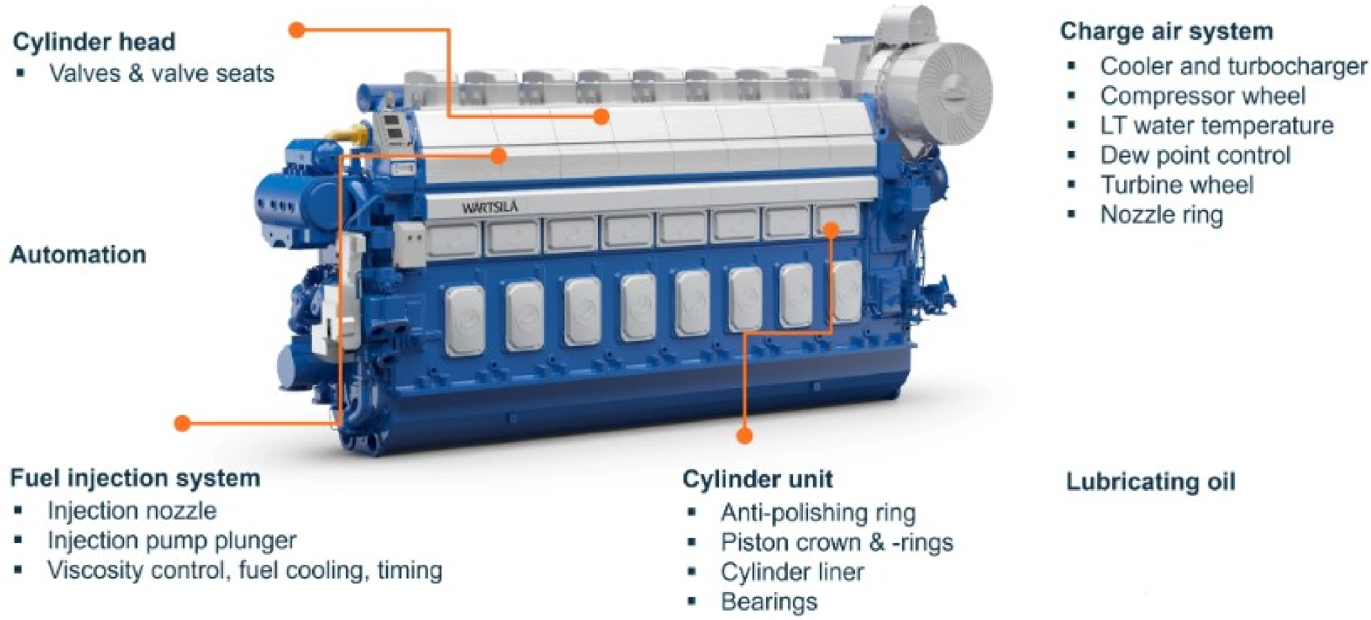


Figure 14. Wärtsilä engine (Bass., 2023).

5.3.2 Implementation of the information model

As OPC foundation provides companion information models for the domain specific use cases, the first step was to examine if some of those can be used. Although there are information models available for many specific domains like mining, process industry and even for commercial kitchen equipments, it turns out that there is not an information model available for shipboard specific systems or for combustion engines. It was also found that many of those companion specifications are using other companion specifications like OPC 10000-100 for Devices and OPC 40001-1 for Machinery as a base, so using of them as the basis of the information model was also investigated in the thesis. However, it was clear that implementation of the information model for engines must be started from scratch.

As the structure of an engine is complex, the challenge was to figure out, what would be the best way for the information model to represent the structure of an engine and its parts including auxiliary components, sensors, and actuators. Some proposal about how the information model for an engine can be constructed have been discussed in Ala-Hynnilä's (2022, p. 72-73) research. As it can be seen in the figure 15, the structure has been divided into smaller groups with the same way that is already used with the UNIC system (Fuel system, Lube oil system, and so on). In the Ala-Hynnilä's suggestion, sensors, and process values of them were connected to physical components of an engine.

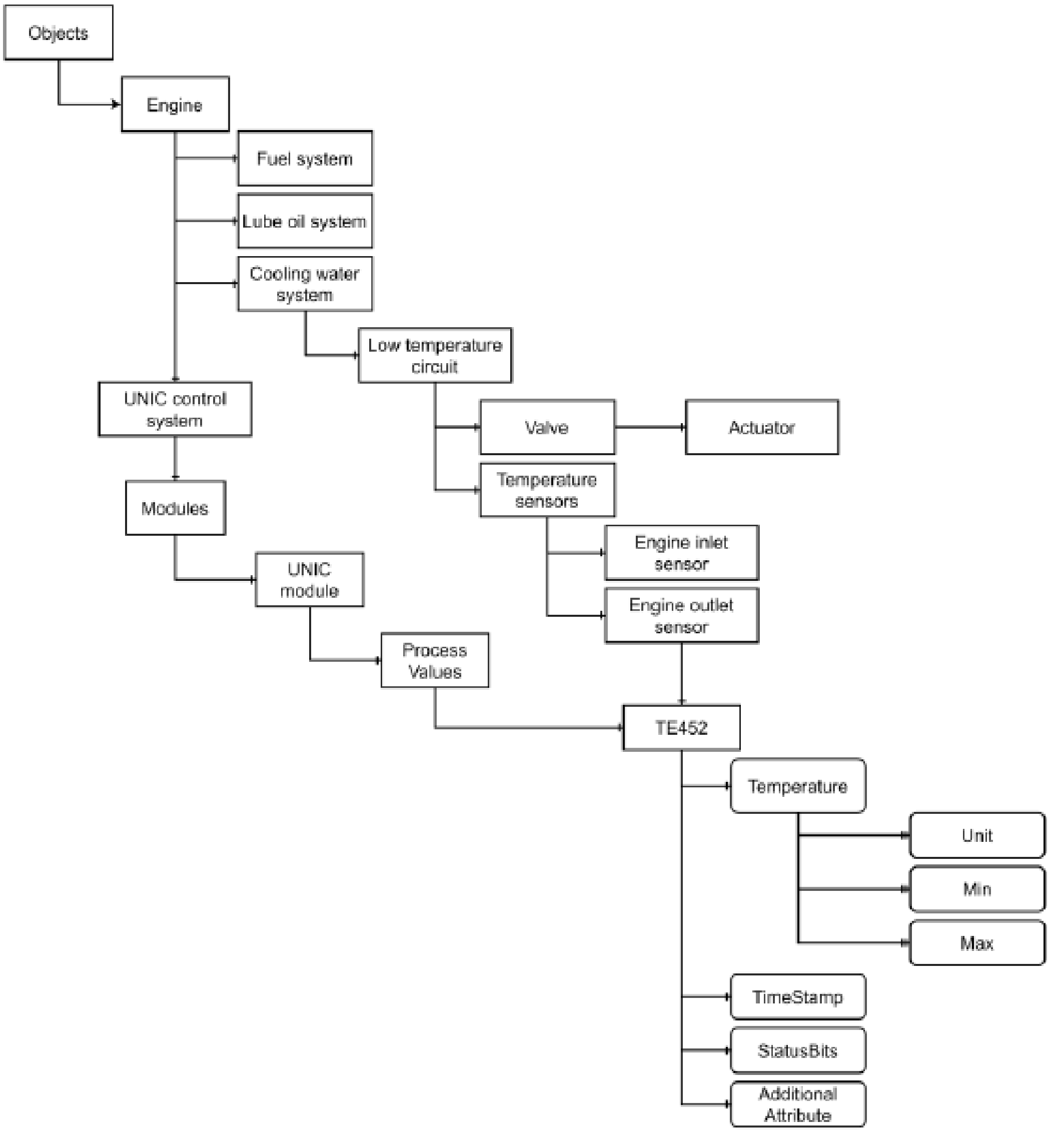


Figure 15. Engine information model suggestion (Ala-Hynnilä, 2022, p. 73).

During the research process of the thesis, it was found that DNV (2023) provides standardized, so called Gmod model to represent all components used shipboard including a reciprocating internal combustion engine. According to Låg et al (2021, p. 6) the Gmod model have been used since 2005 to create unique vessel specific product models, so it is a matured model. According to International Organization for Standardization (ISO, 2018, Annex B) there is also available an alternative model to represent a shipboard data, called jsmea_mac. When compared these two, the Gmod model have much better documentation and is already used for a long time. The Gmod model also provides a hierarchical model of an engine structure and thus the structure can also be applied in the OPC UA information model, while jsmea_mac provides a list of names for different components.

Two different graphical modeling software for the information model creation were tested. The first was SiOME from Siemens and the second was UaModeler from Unified Automation. Both were able be tested without license, but it was not possible to export an information model in xml format from UaModeler without license. The OPC UA core model is built into both modeling software, and it is updated when the software version is updated. It turns out that SiOME is not having the latest built-in OPC UA core model and because of that it was not possible to use the latest companion specifications which are built on top of the latest UA core model. UaModeler was also found to be easier to use and therefore it was used in the thesis.

5.3.3 First version of the information model

According to the information mentioned above, the development of the information model for combustion engines was started according to the DNV's Gmod model. The Gmod model have different versions and the newest in that point was 3-6a. Unfortunately, there was some problems in DNV's web sites since Excel files for the newest version could not be downloaded. Due to that, the second newest version, 3-5a was used. However, the version 3-6a as well as 3-5a can be browsed with a Gmod viewer that DNV provides, and it turns out that in a reciprocating internal combustion engine's point of view they are the same (DNV., 2023). More about the ISO 19848 standard and the Gmod model is discussed in the chapter 3.

When the structure of the Gmod model is examined more closely, it consists of two different kind of building blocks. The first is function types like Asset function, Asset function leaf and Asset function group (DNV., 2023b). The second is equipment types which are physical components of a ship like a steam turbine, a pneumatic power unit and a cooler (DNV., 2023c). However, in this case we were not interested all the types because only an engine is going to be modelled. For modelling of a reciprocating internal combustion engine, types of Product type, Product function group, Product function composition and Product function leaf are needed (DNV., 2023c).

Companion specifications provided by OPC foundation were investigated to see if some of them could be used as a basis for the information model with a structure of the Gmod model. It turns out that nothing exists for this purpose because same kind of a structure is not used elsewhere. According to the above-mentioned information, it was decided to start building of the information model on top of the OPC UA core specification without any additional companion specifications. Two different information models were created according to the structure of the Gmod model, one for basic building blocks of the model and other for a reciprocating internal combustion engine. The idea behind that was to ease the workload in the future if information models for some other shipboard systems manufactured by Wärtsilä would be modeled. The example of the developed information model for combustion engines in that point can be seen in the figure 16.

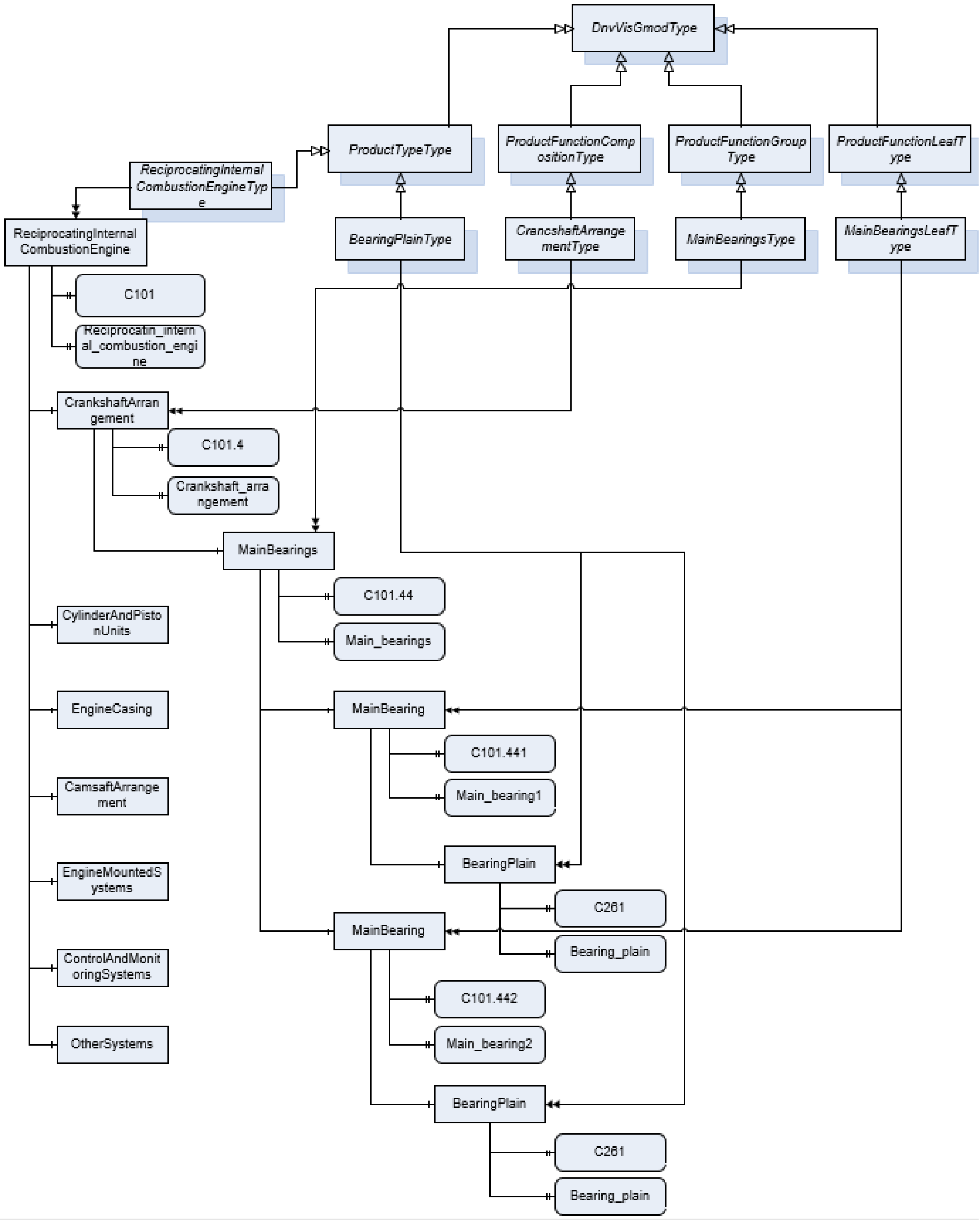


Figure 16. Gmod modelled partially.

The information model was built only partially when its evaluation was performed. In that point it was realized that some parts of the ISO 19848 standard were misunderstood. It was thought that if DNV naming rule is used, it is also necessary to follow the structure of the Gmod model. That is not the case. Instead in point of view of the ISO 19848 standard it eases the creation of the localID for sensors. The Gmod model also contains unnecessary objects in the OPC UA information model point of view. The third thing that was thought about was that it might break the copyright if the Gmod model is just copied to the information model if it is going to be used for commercial purposes. According to these points it was decided to start the implementation of the information model again from the scratch.

At the same time requirements and testing methods were evaluated again. There was a row in the requirement table related to compatibility against to the Gmod model. The removed row can be seen in the table 5. New requirement for the commercial use was added and can be seen in the table 6.

Table 5. Unnecessary requirement.

Title	Priority	Description	Test method
Characteristics of the information model.	Must comply.	Information model shall comply with the DNV's Gmod model described in the ISO19848 standard.	Information model shall be compared to DNV's Gmod model. For comparing SiOME or UAModeler will be used.

Table 6. New requirement.

Title	Priority	Description	Test method
Commercial use.	Nice to have.	It should be possible to commercialize the information model, legal point of view must be kept in mind.	Copyrights must be considered.

5.3.4 Second version of the information model

The development process of the design science is an incremental and iterative and thus it is suitable when several development cycles are needed. As mentioned above, first version of the information model was decided to be rejected completely. The development of the second version was started from implementation of the physical structure of a combustion engine and its mechanical components. The implementation of the feature which will hold the sensor and signal values was decided to be implemented after the physical structure of an engine is modeled.

Creation of the information model again from the beginning was started from examination of companion specifications provided by OPC foundation. As already mentioned, there are no specifications available for marine applications or combustion engines. It was also mentioned earlier that many of the companion specifications are using at least OPC 10000-100 for Devices and possible also OPC 40001-1 for Machinery as a basis and thus, the principles of these specifications have been studied in more details. Inspiration was also got from OPC 40560 for Mining specification as it uses both mentioned specifications as a basis.

The OPC 40001-1 for Machinery provides a solution for the identification of machines and components of machines. The specification uses so called add-in concept that is defined in OPC 10000-3. The idea is to have an identification object for each machine and component in a OPC UA server by using *HasAddin* reference. The same idea was decided to be used also with the engine information model. The other main feature provided by the specification for Machinery is finding of machines and components of machines from a OPC UA server. That was not seen as a necessary feature for the engine information model, and it is also good to keep an information model as simple as possible. Because of that and as the identification object in the specification for Machinery was containing a lot of unnecessary properties that are not needed with the engine information model, it was decided to use only the OPC 10000-100 for Devices as a basis and create own identification objects with the same idea. The identification objects and objects for finding of components and machines provided by the Machinery specification, can be seen in the figure 17.

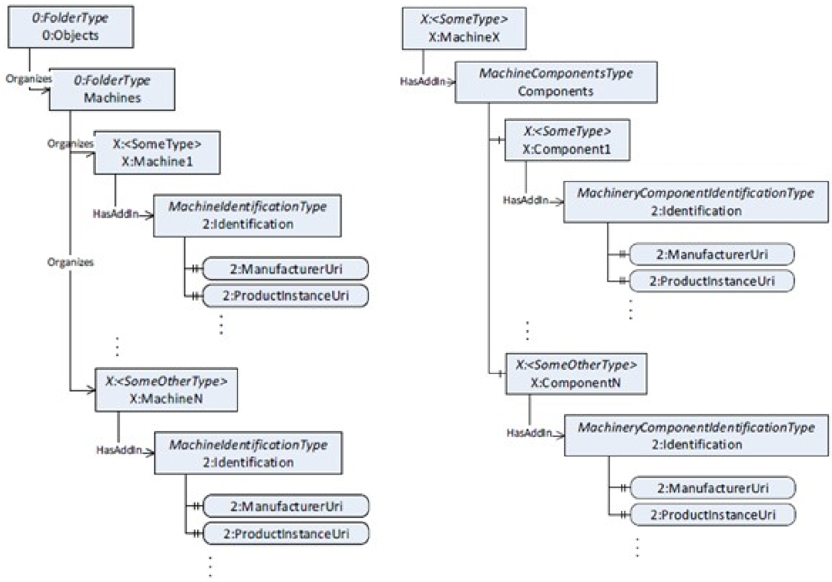


Figure 17. Identification (OPC foundation 2022a, p. 25–28).

According to above mentioned information *CombustionEngineIdentificationType* and *CombustionEngineComponentIdentificationType* were created. These types provide metadata for identification of an engine and components of an engine, such as the name of the manufacturer and the model of the product.

Inspiration and ideas for the object type for a physical combustion engine was got from the specification OPC 40560 for Mining. It uses *TopologyElementType* as a base, which is defined in the specification for Devices as can be seen in the figure 18.

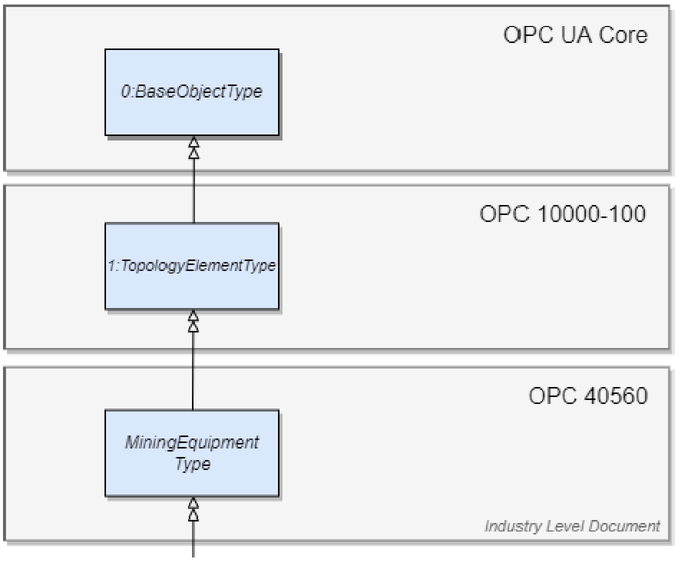


Figure 18. Specification for Mining (OPC Foundation 2022b, p. 16).

According to what was defined in the specification for Mining, *CombustionEngineType* object type was created. In addition of the engine identification object, it also contains several grouping objects which are describing different functional areas and functional compositions of an engine. Components of an engine shall be organized under the grouping objects. Although the structure of the DNV's Gmod model is not used as it was done with the first prototype model, some ideas for the grouping objects was gathered from it. What is done differently compared to the DNV's Gmod model is that there is no need for every component of an engine in the OPC UA information model. For example, there is no need for an object for a crankshaft itself or information what is the type of a crankshaft main bearing. Instead, for example only the object for a main bearing with a temperature sensor or a big end bearing with a temperature sensor are needed and they can be organized under the *CrankshaftArrangement* object.

The structure of the *CombustionEngineType* is defined abstract enough to be used with each engine type and cylinder configuration. Due to the previous, the engine configuration specific components can be added to the engine specific model. With the same idea, components of an engine were created, as an example *TurboChargerType*. The object type for a physical engine can be partially seen in the figure 19.

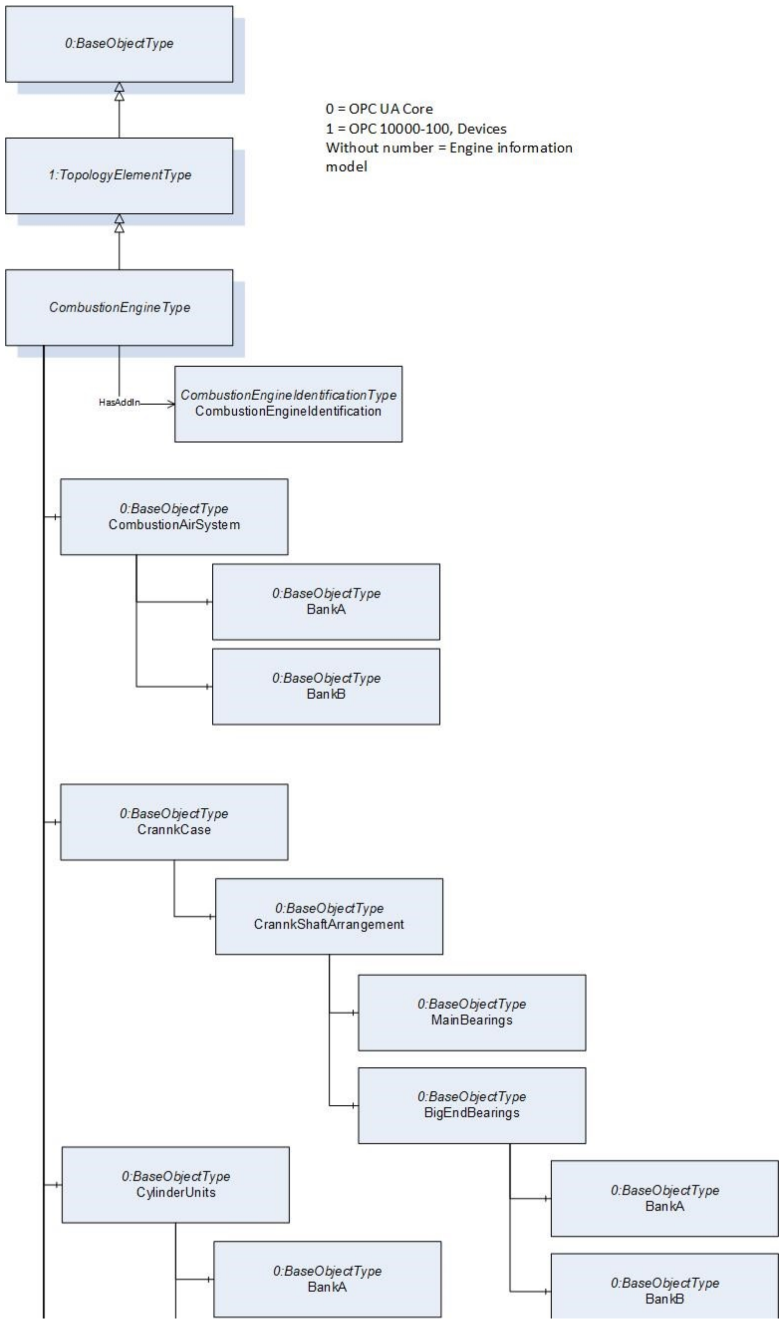


Figure 19. Object type for an engine.

At this point, concept for describing of mechanical components of an engine was defined and concept for describing of a value produced by a sensor or signal is going to be defined next. Inspiration and ideas for them was got from the OPC 30081, UA for PA-DIM and OPC 40001-2: OPC UA for Machinery. In the figure 20 can be seen what is defined related to process values in the specifications. It was explored if the *ProcessValueType* could be used in the information model for combustion engines. Unfortunately, it contains a lot of unnecessary information and thus, it was decided that the object for that purpose will be created from the scratch.

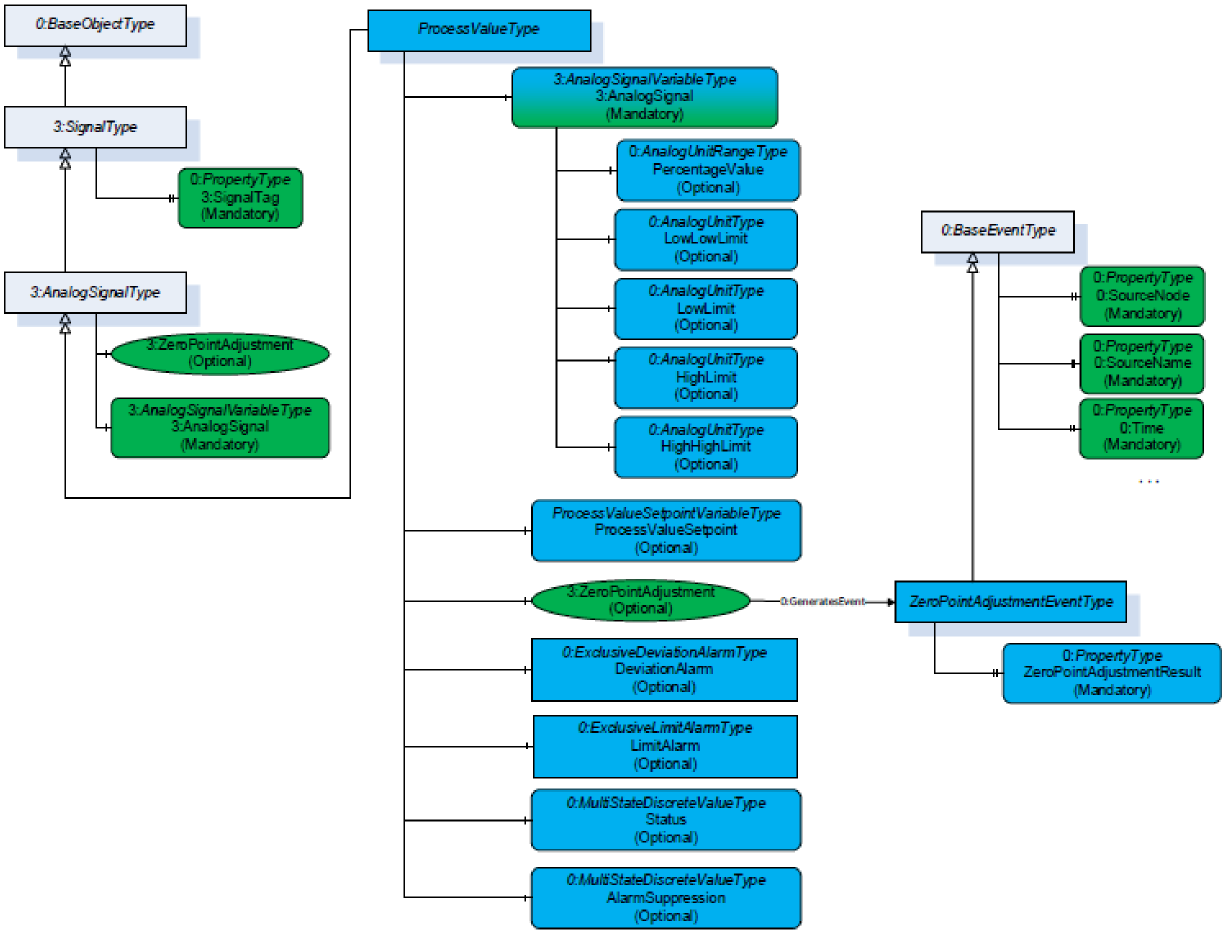


Figure 20. Object for process values (OPC Foundation 2022c, p. 14).

The purpose of the object type for engine signals is to hold a process value or status of the signal and the data related to possible machinery protections of the signal. The other purpose of the object is to hold a metadata for the signal, for example minimum and maximum range of a process value. As it was decided that the information model shall comply with the ISO 19848 standard, the metadata shall be represented in a way that is defined there. One of the most important goals of the standard is to define a standard mechanism for sensor naming and the mechanism should be also included to the information model for engines. According to previous information about the metadata and sensor naming, DataChannelType was created and can be seen in the figure 21. The structure of the object is defined in the standard and thus the structure was used. According to OPC Foundation (2023a, p. 15) it is good to use add-in concept instead of deep type hierarchies whenever possible, and thus it was decided to use the concept with the DataChannelType. DefaultInstanceBrowseName variable was added because it shall be available if the add-in concept is used.

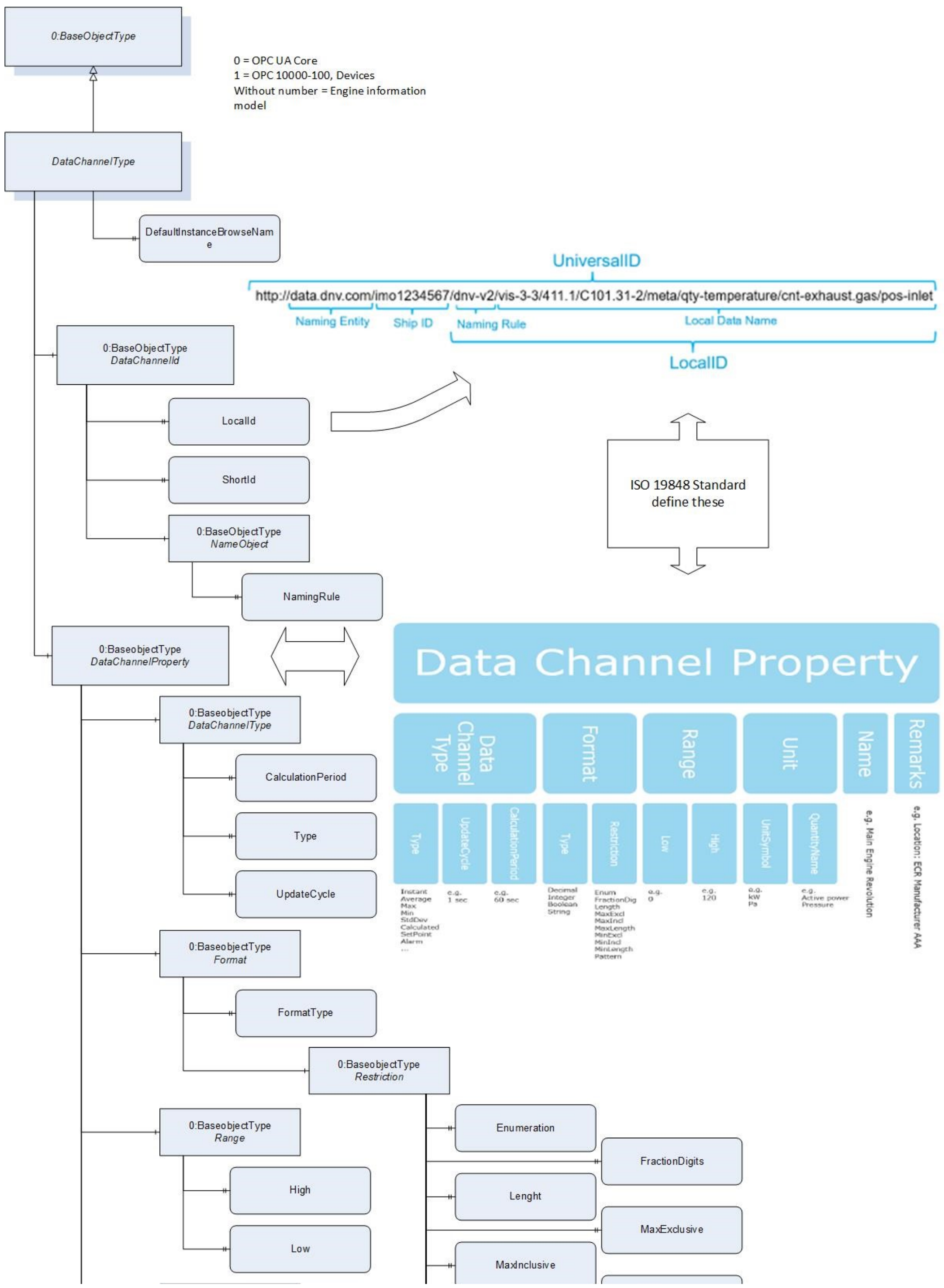


Figure 21. Sensor metadata.

Separate object types were created for a process value, a status information and machinery protections and *DataChannel* object is included to them. Finally, *CombustionEngineSignal-Type* was created and that includes all the types above. The object type includes all the needed for representation of a sensor or signal and can be added where needed. An example of a use case for the type can be seen in the figure 22 where it is representing physical sensors connected to the turbocharger object.

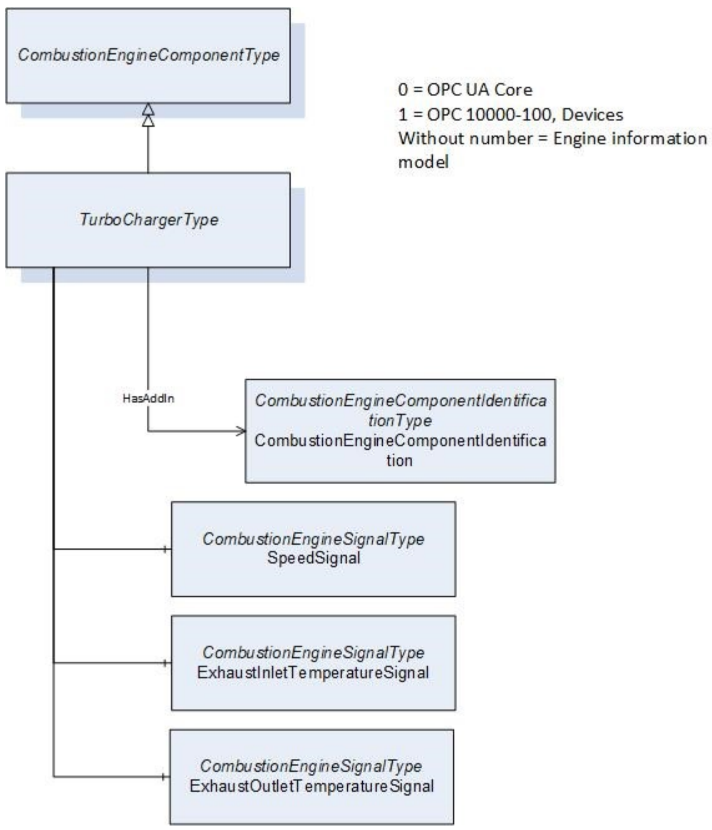


Figure 22. Object for a turbo charger.

5.4 Testing and evaluation

Well defined requirements for the artifact are mandatory in the design science. In addition to just provide requirements, test methods must be defined. Requirements and test methods for the information model were defined before the development of it was started. A table of the requirements and test methods is available in the appendix 1. Some of the requirements were added during the creation of the information model since need for them came up.

Some of the requirements were able to be tested every now and then during the development process, for example validity of the information model since so called consistency check can be done with the UaModeler whenever needed. An example can be seen in the figure 23. Another example was enabling the commercial use of the information model, and it was ensured by creating the information model without copying any other model or the idea.

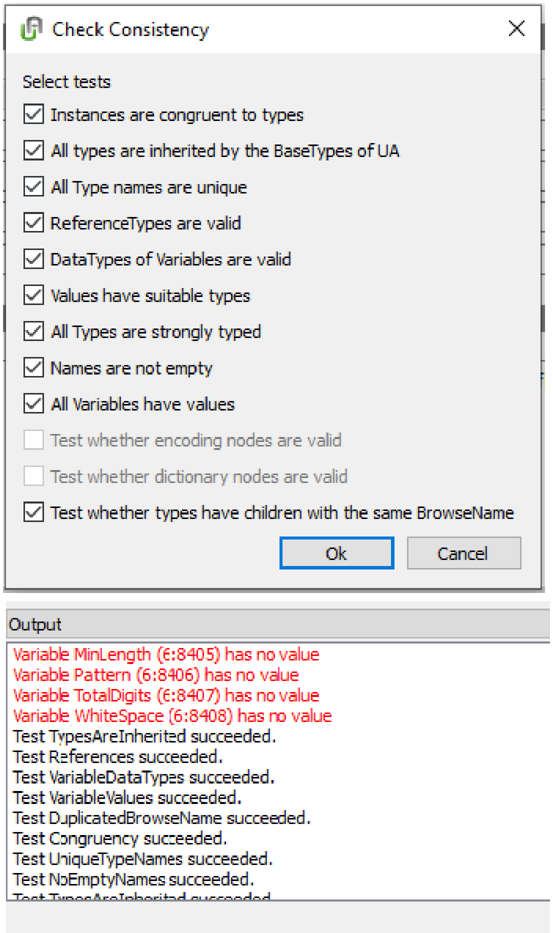


Figure 23. Consistency check.

Some of the requirements were removed during the creation of the information model as they were no longer seen as a valid. An example was requirement for the structure of the information model to comply with the DNV’s Gmod model which was decided to be removed as it is reported within the chapter 5.3.3.

A major part of the requirements and test methods required building of an OPC UA server application with the information model within a Linux environment. For testing purposes, WSL and Visual Studio Code was installed to a PC. WSL is a lightweight Linux environment which can be installed as a Windows subsystem and both environments can be used at the same time (Microsoft, 2022). The Linux environment can be accessed straight from the Visual Studio Code and thus it is easy and fast to test different prototypes. Open62541 stack was used to create server applications for testing purposes. The Open62541 stack is an open source, platform independent implementation of the OPC UA written in a C99 and C++98 languages (Open62541 community, 2022). To browse an information model provided by a server application, a generic OPC UA client application, UaExpert from Unified Automation was used.

When all the needed software was installed, and initial difficulties were overcome, it was quite straight forward to build a server application with the information model. Two different information models were created for testing purposes. The first was the actual information model which includes all needed types to build an engine specific model and the second was an instance of the engine specific example model. Both were exported from UaModeler in a xml format as an open62541 server application is requiring an information model in that format. OPC UA core model is built in to the open62541 stack so a xml model from that is not needed. In addition of the OPC UA core model, the information model for engines was built on top of OPC 10000-100 for Devices and a xml file for that was also needed. Examples from the test environment can be seen in the figure 24 where a server application is running, and it is connected to a client application.

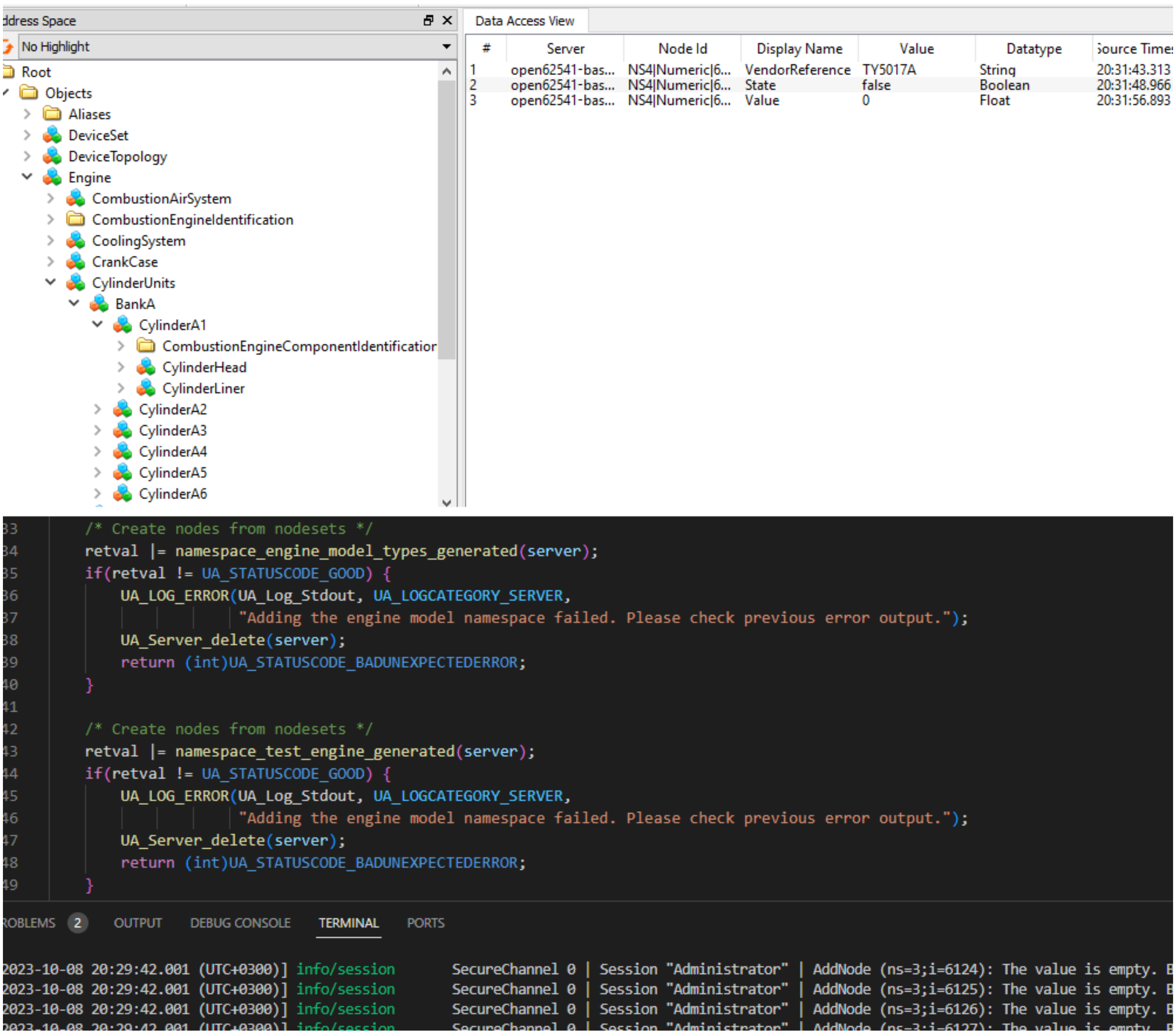


Figure 24. A server connected to a client.

When the OPC UA server application was connected to a client application, main part of requirements were able to be checked manually. Not everything was possible to be tested extensively since the schedule for the thesis was limited. For example, it was not possible to test the information model with all possible engine types and different configurations as can be seen on the table 7. However, a quick walk through against a standard Modbus list data for several different engine types was gone through and it seems that each should be manageable with the information model.

Table 7. Example from requirements.

Title	**Priority**	**Description**	**Test method**	**Test result**
Flexibility of the information model.	Must comply.	The Information model should be flexible, so it can be used with different Wärtsilä engines.	The engine specific information model to be tested with different Wärtsilä engines. It shall be tested with both, a statical model generated manually with SiOME or UaModeler and with an automatically generated model. Manual check.	Partially passed. It was not possible to test all possible engine models and configurations within the thesis schedule. Confirmed with the W32 diesel.

Improvement for the information model was done several times to fulfill requirements. An example is a requirement for the structure of the model, which shall be the same for a client point of view compared to what is showing in the modelling software. The requirement can be seen in the table 8. It was noticed that an optional instance of a variable for one of the machinery protections was showing for a client although it was not showing in the modelling software.

Table 8. Structure of the information model.

Title	Prior-ity	Description	Test method
Structure of the information model.	Must com-ply.	The structure of the infor-mation model shall be identical from a client point of view com-pared to what is showing in the modelling software.	The model showing for a client shall be manually compared to the original.

In the figure 25 can be seen on the left what is showing for a client and on the right, what is showing in a modeling software for the same instance. This mismatch was fixed by creating separate object types for machinery protections with only a status information and another also with a setpoint.

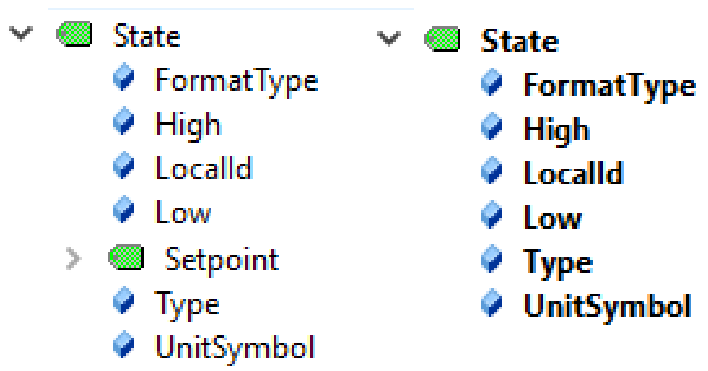


Figure 25. Differences.

One of the requirements was to have a method to automatically create an engine configuration specific information model. However, it was not possible to implement a solution for that within the thesis schedule and thus the requirement was not fulfilled.

6 CONCLUSION AND RESULTS

Design science was used as the research paradigm for the thesis and the guidelines of the paradigm were followed. The research work was started by defining the research problem and indicating the need for a solution to the problem. In this case it was the need for a standardized solution for the communication between different shipboard systems and devices. So, the environment where the problem exists is the whole marine industry. However, within the thesis the research area was limited to the standardization of communication between Wärtsilä's engines and external systems. OPC UA was chosen to solve the problem as it is widely seen as a promising, future proof solution. The focus was the creation of an OPC UA information model for combustion engines.

Since it is important in design science to base the decisions on theoretical knowledge, the thesis project was started by getting familiar with the theories and principles behind OPC UA. It was also researched if any standardized solutions for on-board data exchange exist. During the research process it turned out that the ISO 19848 standard for shipboard data exchange exists. It does not define any rules especially for OPC UA communication but, instead, it provides more general rules about how any information shall be represented and distributed. It was decided that sensor naming rules and the way how sensor metadata is represented according to the standard will be implemented in the OPC UA information model for combustion engines.

Before the development of the information model started, the requirements and test methods for them were defined because well defined requirements are one of the principles of design science. As a development process of design science is iterative, it was suitable for the thesis. The development of the information model was divided into parts. The physical structure of a combustion engine was defined first, to be the frame where mechanical components and sensors can be added. The mechanical parts of an engine were modelled next and lastly, a structure was implemented to enable the presentation of the data produced by a sensor or software. In the beginning of the development work it was hard to figure out suitable structure for object types that represents combustion engine, components of the combustion engine or presentation of a data of the combustion engine and thus several development cycles were performed. The structure of the information model was first developed according to the structure of DNV's Gmod model, but that approach was rejected, and the development was started again from the very beginning.

The result of the thesis is the first development version of the information model, as it was intended to. The model is simplified and thus is not meant to be the final version. The information model represents the physical structure of a combustion engine. In addition to different process values, machinery protection statuses and setpoints, the model also enables information related to the ISO 19848 standard. Most of the requirements which were defined in the beginning were fulfilled. However, evaluation with all different engine types and engine configurations was not possible within the thesis, but since the model was defined to be flexible, it should be possible to apply it with different Wärtsilä engines. Another unfulfilled requirement was the automatic creation of a configuration specific information model which was not implemented. As well it was not possible within the thesis schedule to define ISO 19848 related LocalID’s for all possible signals.

Compared to the current solution with Modbus communication, there are several advantages when OPC UA is used with the information model. For a start, with Modbus it is not possible to use complex information models like the one created during the thesis. Instead, clients need a predefined list of available signals with addresses. With OPC UA there is no need to provide such information in advance since OPC UA clients can browse the information available in the server application. With the information model for combustion engines, also sensor or signal metadata is available in the model, which with Modbus needs to be provided in advance with a manual list. When also the ISO 19848 standard related data is available in the model, it is possible for clients to recognize what is the source of the signal without any manufacturer specific knowledge. These days cyber security is very important and OPC UA has it built in as well as user authentication and a possibility to have different user profiles. User profiles can be used to expose only necessary information for lower-level users while higher-level users can have all possible information available.

For the future development, the implementation of the automatic creation of engine configuration specific information model shall be created, since it was not possible within the thesis schedule. The proof of concept is already available for one possible method. The method builds an information model based on the configuration file and xml model that contains the types which are used as building blocks. There were also some other development ideas for the future: several more specific sub types of the object type for engine signals could be added. For example, there could be object types for some specific temperature sensors like

PT100. As well, there could be sub types for the object types of mechanical components like turbo chargers with details of different manufacturers or models.

7 FINAL WORDS

The topic of the thesis was very interesting and topical. OPC UA is a very broad collection of standards with many details and thus its principles are not easy to understand right away, especially if one does not have much experience on automation and software development, like me. My daily work is not software development and thus this kind of tasks are not my area of expertise. Because of that, the basic things related to software development were time consuming in the beginning. The most time was spent studying the principles and details of OPC UA.

However, the first version of the information model was finished. I also learned a lot about OPC UA and software development and thus, I feel this was worth all the effort. I want to thank Gustavo Ramirez from Wärtsilä who helped me a lot with the practical things. I also want to thank Timo Koskiahde from Wärtsilä and Ismo Tupamäki from SeAMK. Special thanks to Petra Autio for patience when I was writing the thesis in many evenings.

BIBLIOGRAPHY

Ala-Hynnilä, M. (2022). *Replacing internal communication protocol in UNIC control system.* https://osuva.uwasa.fi/handle/10024/14484

Bass, C. (15.01.2023). Want to cut costs and extend CII compliance? The top 4 ways good maintenance practices can help. https://www.wartsila.com/insights/article/cut-costs-and-extend-cii-compliance-with-maintenance?utm_source=linkedin&utm_medium=social-org&utm_term=corp&utm_content=in-ciiservice&utm_campaign=in-mp

Bauer, H., Dietche, K., Crepin, J & Dinkler, F (2003). Autoteknillinen taskukirja. Robert Bosch GmbH (6.p.).

Damm, M (15.4.2014). Introduction to OPC UA. https://www.unified-automation.com/downloads/webinars/introduction-migration-to-opc-ua.html

DNV. (2023). *Gmod viewer.* https://vista.dnv.com/gmod

DNV. (2023b). Vessel VIS 3-5a.xlsx. https://vista.dnv.com/static/v3-5a/Vessel%20VIS%203-5a.xlsx

DNV. (2023c). Products VIS 3-5a.xlsx. https://vista.dnv.com/static/v3-5a/Products%20VIS%203-5a.xlsx

Hevner, A., Salvatore, T., & Park,J. (March, 2004). Design Science In Information System Research. MIS Quarterly Vol 28(1). https://www.researchgate.net/publication/201168946_Design_Science_in_Information_Systems_Research

Hoppe, S. (June, 2017). *There Is No Industrie 4.0 without OPC UA.* https://opcconnect.opcfoundation.org/2017/06/there-is-no-industrie-4-0-without-opc-ua/

Industry40tv. (02.09.2020). What is OPC UA and How it Works?. https://www.youtube.com/watch?v=vRk42W_4R0o&list=PLIrJJXPVFRvjHsA9tta8yULOB8nPUO_G7&index=1

International Organization for Standardization (ISO). (2018). *Ships and marine technology: Standard data for shipboard machinery and equipment. (ISO 19848:2018).* https://www.iso.org/standard/66406.html

Jokinen, T (19.02.2021). Konstruktiivinen tapaustutkimus ja suunnittelutiede – kaksi insinööritieteisiin soveltuvaa tutkimusotetta. https://blogi.oamk.fi/2021/02/19/konstruktiivinen-tapaustutkimus-ja-suunnittelutiede-kaksi-insinooritieteisiin-soveltuvaa-tutkimusotetta/

Järvinen, P. (February, 2006). Onko innovaatioiden suunnittelu tiedettä?. Systeemityö-lehti. https://www.pcuf.fi/sytyke/lehti/kirj/st20062/ST062-25A.pdf

Kominek, D. (October, 2017). *Should I Use OPC UA or MQTT or AMQP?.* https://opcconnect.opcfoundation.org/2017/10/should-i-use-opc-ua-mqtt-amqp/

Låg, S., Vindøy, V., Ramsrud, K. (2021). *A Standardized Sensor Naming Method to Support Digital Twins and Enabling New Data Driven Applications in the Maritime Industry.* https://vista.dnv.com/static/Hiper2021_Standardized_sensor_naming_method_paper.pdf

Macgregor. (November, 2017). Maritime Data Engine. https://www.macgregor.cn/globalassets/picturepark/imported-assets/83864.pdf

Mahnke, W., Leitner, S., & Damm, M (2009). *OPC Unified Architecture.* Springer.

Microsoft. (27.06.2022). Windows Subsystem for Linux: Documentation. https://learn.microsoft.com/en-us/windows/wsl/

OPC foundation. (01.05.2022a). *OPC UA for Machinery: Part 1: Basic Building Blocks.* https://opcfoundation.org/developer-tools/documents/view/213

OPC foundation. (01.09.2022b). *OPC UA for Mining - General.* https://opcfoundation.org/developer-tools/documents/view/249

OPC foundation. (11.01.2022c). *OPC UA for Machinery: Part2: Process Values.* https://opcfoundation.org/developer-tools/documents/view/315

OPC foundation. (16.01.2023a). *OPC Unified Architecture: UA Modelling Best Practices.* https://opcfoundation.org/developer-tools/documents/view/312

OPC foundation. (n.d.). *What is OPC?*. https://opcfoundation.org/about/what-is-opc/

OPC foundation. (2023b). *UA specifications.* https://opcfoundation.org/developer-tools/documents/?type=Specification

OPC foundation. (21.02.2022d). *OPC Unified architecture: Part 6: Mappings.* https://opcfoundation.org/developer-tools/documents/view/163

OPC foundation. (21.02.2022e). *OPC Unified architecture: Part 3: Address Space Model.* https://opcfoundation.org/developer-tools/documents/view/160

OPC foundation. (22.11.2017). *OPC Unified architecture: Part 1: Overview and Concepts.* https://opcfoundation.org/developer-tools/specifications-unified-architecture/part-1-overview-and-concepts/

Open62541 community. (04.08.2022). open62541 Documentation. https://www.open62541.org/doc/1.3/

Peffers, K., Tuunanen, T., & Rothenberger, A. (January, 2007). A design science research methodology for information systems research. Journal of Management Information Systems. https://www.researchgate.net/publication/284503626_A_design_science_research_methodology_for_information_systems_research

Phoenix contact. (2023). Maritime 4.0-the digital transformation of the maritime industry. https://www.phoenixcontact.com/en-pc/industries/marine-and-offshore/maritime-4-0#ex-8ppy3

Pipero, C. (24.7.2018). *OPC UA Information Models.* https://www.youtube.com/watch?v=BBktfFBUEjI

Südekum, N. (02.07.2020). Shipboard Automation Systems Skim Ahead of the Competition. https://www.wago.com/global/digitization/adaptability/shipboard-automation-systems-skim-ahead-of-the-competition

Südekum, N. (16.08.2018). Efficient System Integration Through Service-Oriented Software in Automation. https://www.wago.com/global/marine-offshore-solutions/alarm-monitoring/dima-efficient-system-integration

Turner, A. (05.06.2020). UEI Technical Master Class: OPC-UA & Modbus. https://www.youtube.com/watch?v=oNEsNWKYG0o

Unified Automation. (22.12.2022a). OPC UA Fundamentals. https://documentation.unified-automation.com/uasdkhp/1.7.0/html/_l1_opc_ua_fundamentals.html

Unified Automation. (22.12.2022b). OPC Unified Architecture Overview. https://documentation.unified-automation.com/uasdkhp/1.7.0/html/_l2_opc_ua_fundamentals_overview.html#OpcUaFundamentalsMetaModel

Unified Automation. (22.12.2022c). OPC UA Software Layers. https://documentation.unified-automation.com/uasdkhp/1.7.0/html/_l2_opc_ua_software_layers.html

VDMA. (14.03.2023). Improved system integration on board with maritime MTP standards. https://www.vdma.org/viewer/-/v2article/render/51523662

Wago. (31.01.2020). On-Board Technology is Changing. https://www.wago.com/global/digitization/on-board-technology-is-changing

Wärtsilä. (2023). Wärtsilä businesses in brief. https://www.wartsila.com/media/businesses-in-brief

APPENDICES

Appendix 1. Requirements for the information model

Appendix 1. Requirements for the information model

Title	Prior-ity	Description	Test method	Test result
Characteristics of the infor-mation model.	Must com-ply.	The Information model shall describe the structure of a physical Wärtsilä engine. The same information that is available with the current Modbus communication shall be provided.	The Information model shall be com-pared to physical Wärtsilä engines. For comparing SiOME or UaModeler will be used. The Information inside the model shall be compared against the current Modbus lists. Manual check.	Passed. The infor-mation model browsed by UaModeler de-scribes structure of physical engines. All the Modbus list related information can be added.
Flexibility of the information model.	Must com-ply.	The Information model should be flexible, so it can be used with differ-ent Wärtsilä engines.	The engine specific information model to be tested with different Wärtsilä en-gines. It shall be tested with both, a statical model generated manually with SiOME or UaModeler and with an automatically generated model. Manual check.	Partially passed. It was not possible to test all possible engine models and configurations within the thesis sched-ule. Confirmed with the W32 diesel.

Title	Prior-ity	Description	Test method	Test result
OPC UA server with information model.	Must com-ply.	It shall be possible to generate OPC UA server application with the infor-mation model by using the Open62541 stack.	The server application to be gener-ated with the information model by using the Open62541 stack.	Passed. The OPC UA server application build successfully with the in-formation model by us-ing the Open62541 stack.
Automatic crea-tion of the en-gine specific in-formation model.	Must com-ply.	It should be possible to generate an engine specific information model automatically.	An engine specific information model to be generated automatically for dif-ferent Wärtsilä engines.	Failed. Has not been implemented.
Validity of the in-formation model.	Must com-ply.	The information model shall be com-patible with OPC UA rules.	The information model shall be tested and pass the validation tests of Si-OME or UaModeler.	Passed. No errors on UAModeler consistency check.

Title	Prior-ity	Description	Test method	Test result
Browsing of the information model.	Must com-ply.	It must be possible for a generic OPC UA client to browse the infor-mation model from a OPC UA server application.	A server application to be generated with the information model by using the Open62541 stack. The infor-mation model will be browsed with the generic OPC UA client, UaExpert. Manual check from a client applica-tion.	Passed. A server appli-cation with the infor-mation model ran in WSL environment. UaExpert client appli-cation connected to the server and confirmed that all the information is available.
Reading and writing of pro-cess values.	Must com-ply.	It must be possible for a generic OPC UA client to read sensor values from a OPC UA server and in some cases also write them.	A server application will be browsed with the UaExperts and sensor values will be read and also written if needed. Manual check from a client application.	Passed. Several ran-domly picked sensor values, read, and writ-ten successfully. Also confirmed that it is not possible to write when access level is read only.

Title	Priority	Description	Test method	Test result
Reading the statuses and setpoints of machinery protections.	Must comply.	It must be possible for a generic OPC UA client to read statuses and setpoints of machinery protections from a OPC UA server.	The server application will be browsed with the UaExperts and statuses and setpoints of machinery protections will be read. Manual check from a client application.	Passed. Several randomly picked machinery protection statuses and setpoints read successfully. Also confirmed that it is not possible to write when access level is read only.
Structure of the information model.	Must comply.	The structure of the information model shall be identical from a client point of view compared to what is showing in the modelling software.	The model showing for a client shall be manually compared to the original.	Passed. Several randomly picked parts of the model compared manually and confirmed to have the same structure.
Local ID.	Must comply.	It should be possible to represent sensor naming by LocalID, defined in ISO 19848 standard.	When some specific sensor information is browsed from the information model provided by a OPC UA server, it must be possible for a client to read sensor naming data in a form of localID. Manual check from a client application.	Passed. Local IDs added manually for signals in the information model. Connected UaExperts client to the server application and local ID's read successfully.

Title	Prior-ity	Description	Test method	Test result
Sensor metadata	Must com-ply.	A sensor metadata should be repre-sented according to ISO 19848 standard.	When some specific sensor infor-mation is browsed from the infor-mation model provided by a OPC UA server, it must be possible for a client to browse a sensor metadata related to ISO 198484 standard. Manual check from a client application.	Passed. The meta data of sensors or signals added manually in the information model. Connected UaExperts client to the server ap-plication and metadata read successfully from several randomly picked signals.
Commercial use.	Nice to have.	It should be possible to commercial-ize the information model, legal point of view must be kept in mind.	Copyrights must be considered.	Passed. Completely own model created.