UNIVERSITY OF VAASA

FACULTY OF TECHNOLOGY

ELECTRICAL ENGINEERING

Johan Audas

REQUIREMENT ANALYSIS OF A SIGNAL DATABASE FOR POWER PLANT AUTOMATION SIGNALS

Master's thesis for the degree of Master of Science in Technology submitted for inspection, Vaasa, 10th of September 2014.

Supervisor	Prof. Kimmo Kauhaniemi
Instructor	B.Sc. Niklas Svevar
Examiner	Prof. Jouni Lampinen

FOREWORDS

Although data of companies has been handled in digital information systems for a long time already, the information systems start to be outdated in many cases. The availability of design-, requirements- and technical specifications-data has grown so much that the older information systems cannot be efficiently used anymore because of the large amount of manual work. Quality requirements also create a need for better traceability and revision control, which often are problems with the older information systems. Because of this, more modern information systems based on object-oriented thinking and pre-defined workflows are taken into use to meet the new data requirements. This thesis is a base for the development of a new database solution for design data of power plant automation and control signals.

I would like to thank my instructor and colleague Niklas Svevar from Wärtsilä for the idea for this thesis and for support during the writing. I also thank Niklas Wägar and Timo Rätti from Wärtsilä for the chance to flexibly work with this thesis during the last year. A thank you also goes to all the people who have participated in the meetings where this topic has been discussed and requirements have been gathered. Finally I would like to thank professors Kimmo Kauhaniemi and Jouni Lampinen for their instructions during the writing of this thesis.

CONTENTS

SYMBOLS AND ABBREVIATIONS

API	Application Programming Interface
AVR	Automatic Voltage Regulation
CAD	Computer Aided Design
CHP	Combined Heat and Power
CPU	Central Processing Unit
DC	Direct Current
DCS	Distributed Control System
ER diagram	Entity-Relationship diagram
ERP	Enterprise Resource Planning
GUI	Graphical User Interface
I/O	Input/Output
IDM	Internal Document Management
LNG	Liquefied Natural Gas
LV	Low Voltage
MoPo-list	Monitoring Point-list
MV	Medium Voltage
OOSAD	Object-Oriented Systems Analysis and Design
PaaS	Platform as a Service
PDM	Product Data Management
PE	Project Engineer
PLC	Programmable Logic Controller
PLM	Product Lifecycle Management
RAD	Rapid Application Development
RTU	Remote Terminal Unit
RUP	Rational Unified Process
SCADA	Supervisory Control and Data Acquisition
TCP/IP	Transmission Control Protocol / Internet Protocol
UML	Unified Modeling Language
WISE	Wärtsilä Information System Environment
WOIS	Wärtsilä Operator’s Interface System
XML	Extensible Markup Language

VAASAN YLIOPISTO
Teknillinen tiedekunta

Tekijä:	Johan Audas
Diplomityön nimi:	Voimalaitosten automaatiosignaalien tietokannan vaatimusten analysointi
Valvoja:	Professori Kimmo Kauhaniemi
Ohjaaja:	Insinööri Niklas Svevar
Tarkastaja:	Professori Jouni Lampinen
Tutkinto:	Diplomi-insinööri
Oppiaine:	Sähkötekniikka
Opintojen aloitusvuosi:	2008
Diplomityön valmistumisvuosi:	2014 **Sivumäärä:** 74

TIIVISTELMÄ

Signaalilistat joita käytetään Wärtsilä Power Plantsin säätö- ja ohjausjärjestelmien suunnitteluun tehdään tänä päivänä taulukkolaskentaohjelmisto-pohjaisella määrittelytyökalulla. Tämän työkalun ongelmia ovat huono ylläpidettävyys, iso manuaalinen työmäärä sekä signaalitietojen jäljitettävyys. Ylläpito-ongelmat johtuvat kasvaneista voimalaitos-koosta, kovemmista asiakasvaatimuksista ja siitä että signaalitiedot löytyvät useista eri taulukkolaskentaohjelmisto-tiedostosta. Näiden ongelmien ratkaisemiseksi on suunniteltu uutta signaalitietokantaa. Tässä työssä tietokannan vaatimukset on analysoitu, tietokannasta on tehty toiminnalliset kuvaukset sekä luotu korkean tason malli tietokannasta.

Pääasialliset osalliset signaalitietokantaan liittyen on tunnistettu ja heidät on haastateltu vaatimusten keräämiseksi sekä nykyisten signaalilistojen luontiin liittyvien ongelmien tunnistamiseksi. Kehitystyö tehtiin läheisessä yhteistyössä Wärtsilä Power Plantsin suunnitteluosaston, projektijohdon ja säätöjärjestelmien toimittajien kanssa. Teoriaosuudessa on esitelty poimintoja kirjallisuudesta tietokannansuunnitteluun ja vaatimustenhallintaan liittyen. Tätä teoriaa on käytetty tietokantamallin määrittelyyn sekä vaatimusten tunnistamiseen tietokannan kehitysvaiheen alkuosassa.

Päämääränä oli kerätä tärkeimmät vaatimukset, luoda ratkaisuja nykyisille ongelmille ja tunnistaa mahdollisia ratkaisuja tietokantaa varten. Tulokset näyttävät että on olemassa suuri määrä signaalitietoa joka on hajautettuna eri paikoissa. Erilaisia syitä nykyisten projektipuolen ongelmiin signaaleihin liittyen tunnistettiin ja päävaatimukset suunnitellulle tietokannalle kerättiin. Työssä mallinnettiin tietokannan suunnitellut toiminnallisuudet ja esitettiin mahdollisia ratkaisuja signaalimäärittelytyöhön. Uusia löydöksiä tietokantayhteyksistä sekä signaalitietojen elinkaarenhallinnasta esitettiin tulevaisuuden signaalimäärittelytyön vaatimuksia varten.

AVAINSANAT: voimalaitos, signaali, tietokanta, automaatio

UNIVERSITY OF VAASA		
Faculty of Technology		
Author:	Johan Audas	
Topic of the Thesis:	Requirement Analysis of a Signal Database for Power Plant Automation Signals	
Supervisor:	Prof. Kimmo Kauhaniemi	
Instructor:	B.Sc. Niklas Svevar	
Examiner:	Prof. Jouni Lampinen	
Degree:	Master of Science in Technology	
Major of Subject:	Electrical Engineering	
Year of Entering the University:	2008	
Year of Completing the Thesis:	2014	**Pages:** 74

ABSTRACT

The signal lists used for the control and automation system design of Wärtsilä Power Plants are today made with a spreadsheet software-based configuration tool. The problems with this tool are difficult data maintenance, a large manual workload and traceability of the data of configured signal lists. The maintenance problems are caused by larger power plant sizes, stricter customer requirements and scattered information in different spreadsheet software-files. To solve these problems, a database solution is planned to be developed. In this thesis, the requirements of the database are analyzed, functional descriptions are made and a high-level model of the database solution is created.

The main stakeholders of the database system have been identified and interviews have been held with all of them to gather requirements and get information of current problems related to the creation of the signal lists. The development work is done in close cooperation with the Wärtsilä Power Plants R&D department, project management and control system suppliers. Theory of database design and requirements management is collected from scientific literature. This theory is used to identify requirements in the early phases of the database development and in the creation of a conceptual model of the database.

The goal was to collect the main requirements, create solutions for current problems and to identify possible solutions for the planned database. The results show the large amount of signal data that is today scattered in different sources. Reasons for problems in power plant project work and control system design have been identified and the main requirements for the database solution have been classified. The planned database functionalities are modelled and possible solutions to fulfill the signal list configuration work are presented. New findings of connectivity to other information systems and lifecycle management of the signal design are brought up to meet future requirements of the power plant signal configuration work.

KEYWORDS: Power Plant, Signal, Database, Automation

1 INTRODUCTION

Enterprises are today moving more and more towards centralized product data management by using PDM (Product Data Management) software. This means that all information about products, including drawings, costs and requirements, is handled by PDM software. Wärtsilä is already today integrating the whole customer delivery process to PDM systems. All data from the sales quote all the way to the end of the power plant lifecycle, is planned to be handled with PDM software. The goal is to get rid of today's manual document handling and get all data into one information handling system.

Like most of the design, sales, customer relation and requirement data, signal data for the signal set-up of power plants has been handled manually in spreadsheet-files with Microsoft Excel. These signals are monitoring and control signals for different power plant processes. Wärtsilä offers engine-powered power plant solutions which include besides the engines, a lot of auxiliary equipment like pumps, fuel separators, tanks and switchgear. All these different systems are connected to the power plant automation and control center.

Because of the large variation in different power plant solutions, the signal setup varies a lot and there is no standard signal connection setup. The manual workload to provide the right connections for hundreds of signals has been large. The current way is to use a spreadsheet-file for making the configuration but this tool starts to be difficult to maintain. In this thesis a database solution for handling the signals is investigated. So far the signal list has been configured with Microsoft Excel-macros but the aim is to pull the project scope from information systems where the scope is already configured and generate scope specific signal lists.

Besides generating a project specific signal list, the maintenance of the signal data also has to be considered. The R&D department needs to be able to easily update and maintain the signal data for ensuring up-to-date data. This creates requirements on the user interface for the database management.

However, before going into generation of project specific signal lists and database maintenance, a model for the database needs to be developed. This model includes descriptions of the signal data and its connections to parts. The attributes of the signals also need to be mapped. By using UML (Unified Modeling Language) or ER (Entity-Relationship) -diagrams, a model of the database can be created. At the moment there are several lists of signals and they need to be connected in some way to get a clear picture of how the complete signal set-up for power plants is made. This is also essential when designing a database for power plant signals. It needs to be known how the database should be connected to for example engine signals, which have their own signal list.

The methodology of this study is to gather and analyze requirements of the database by interviewing different stakeholders. These include the R&D department, project engineers and control system suppliers. The interviews are focused on how they use the signal list, what data is important, what problems there are today and what kinds of requirements they have for the planned database system.

Before going into the database solution, the basics of power plant control and automation systems will be presented. After that, theory about PLM (Product Lifecycle Management), information system development and UML will be reviewed. This Master's thesis is written as an assignment by Wärtsilä Power Plants and it is done while working at Wärtsilä in the Power Plants Product Data Management team.

2 DESCRIPTION OF CURRENT SYSTEMS

2.1 Power plant electrical and automation systems

This section will describe the electrical and automation systems of a power plant in a general level. The spreadsheet-based tool used for making the signal configuration is called the Monitoring Point list or the MoPo-list. The signals in the MoPo-list are different monitoring and control signals for these automation systems.

2.1.1 Description of Wärtsilä power plants

Wärtsilä Power Plants provide combustion engine powered plants for power generation in the 3–600 MW range. The engines run on natural gas, LNG (Liquefied Natural Gas), biogas, heavy fuel oil, light fuel oil or liquid bio fuels (Power Plants Solutions 2013). Wärtsilä can provide the complete power plant solution which includes besides the power generation set, automation and control systems, tanks, buildings, switchyard, CHP (Combined Heat & Power) - modules and other auxiliary systems.

The power plants are built of modules which have standardized interfaces to each other. The main parts of a power plant are:

- Engine
- Generator
- MV (Medium Voltage) system
- LV (Low Voltage) system
- DC (Direct Current) system
- Control systems
- Auxiliary system
- Transformers

- Switchyard
- Radiators

As shown in the following figure, the plant site has several different units which need to be connected to the plant automation and monitoring system.

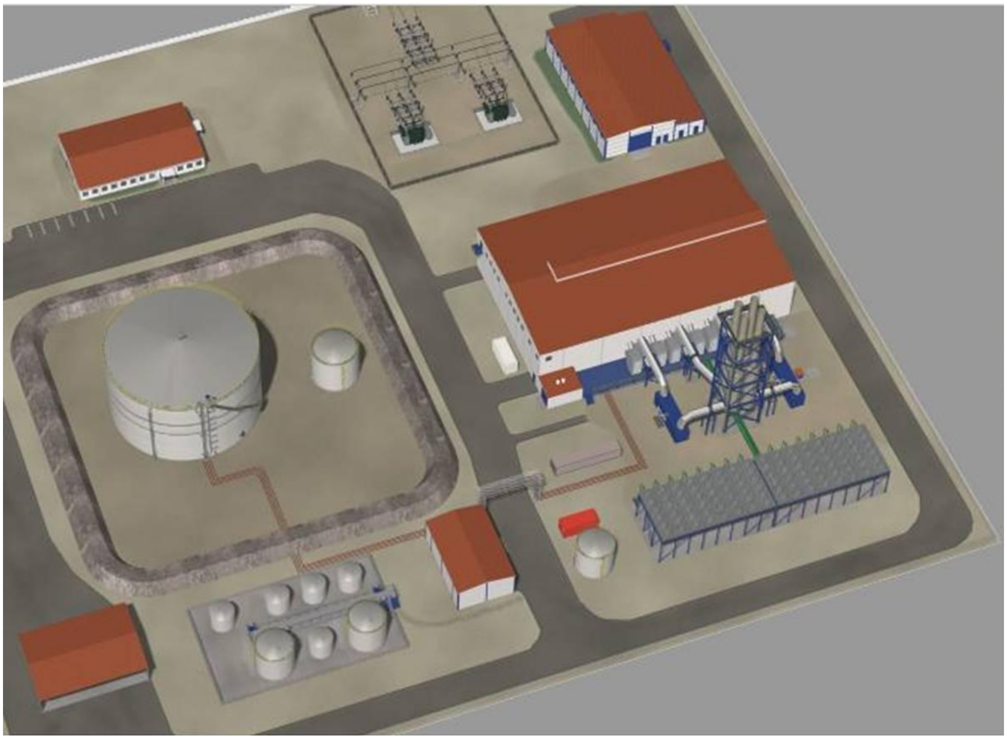

Figure 1. 3D-model of a typical power plant (Wärtsilä ELWIS (Electrical Wisdom) 2013).

2.1.2 Plant automation

The signals in the MoPo-list are control and automation signals. This sub-section describes the most common automation systems in Wärtsilä power plants. Figure 2 shows the main electrical parts and their connections to each other.

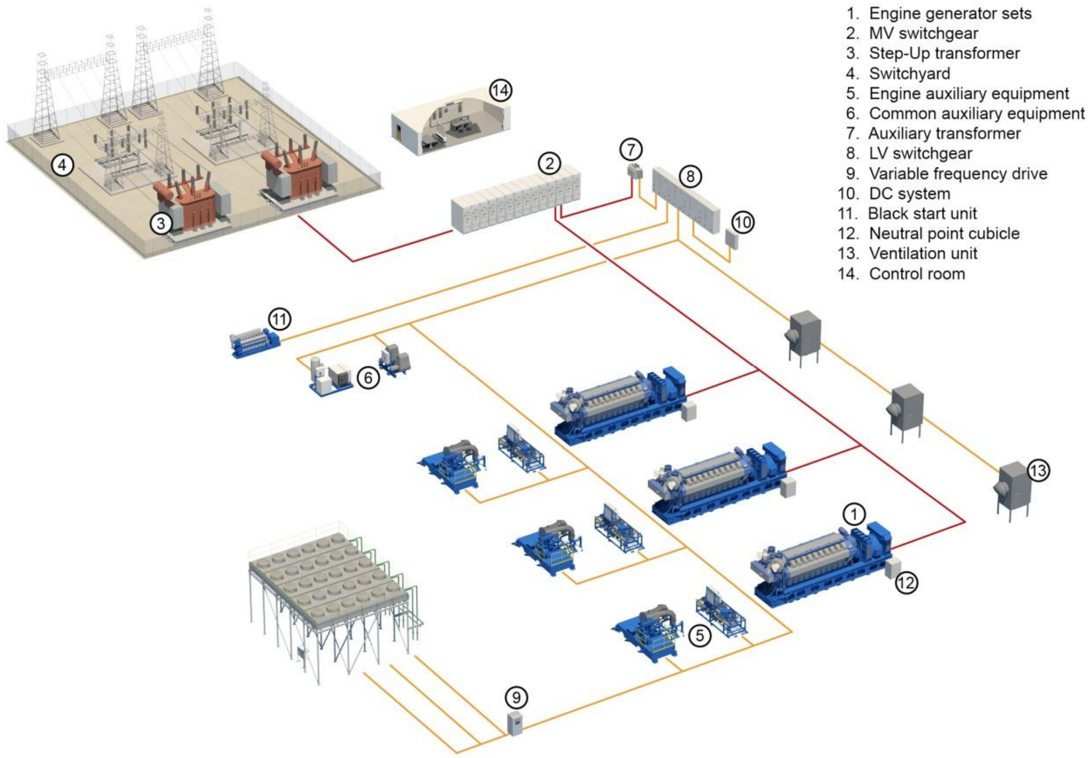

Figure 2. Overview of the power plant electrical systems (Wärtsilä ELWIS (Electrical Wisdom) 2013).

The engines have automated control systems which control for example engine speed and power. The engine control is a two-way automation system. The most important measurements from the engine are speed, lube oil pressure, cooling temperature, bearing temperatures and exhaust gas temperatures. The most important input data to the engine is generator load, grid synchronization and generator breaker status. (Wärtsilä ELWIS (Electrical Wisdom) 2013.)

Besides the generating sets, there are many other modules that are connected to the plant automation and control system. The plant site has several tanks for fuel, lube oil, used lube oil, fire water and so on. These all have level indicators and most of them have alarms for high or low levels. In addition, temperatures and flows might need to be measured. These measurement and control signals are connected to the plant PLCs

(Programmable Logic Controllers). From the PLCs the signals are further communicated to the WOIS (Wärtsilä Operator's Interface System) for visualization.

2.1.3 Main control system

The three basic tasks of the control system are supervision, protection and control. Protection is done by relays and the UNIC engine control. UNIC is a trademark of Wärtsilä and the name comes from Wärtsilä unified control. The UNIC takes care of engine protection by measuring temperatures, pressures and speed, among other things. The electrical system is protected by relays, for example for the generator, transformer and switchyard. Control is done by the UNIC, AVR (Automatic Voltage Regulation) and PLCs. The PLCs control all auxiliary modules and in addition, they also control the UNICs and AVRs. (Wärtsilä ELWIS (Electrical Wisdom) 2013.)

For supervision, the Wärtsilä Operator's Interface System (WOIS) is used. This system is normally located in a separate building at the power plant. The WOIS functions as a SCADA (Supervisory Control and Data Acquisition) for Wärtsilä. The WOIS has a graphical user interface where plant operators can see all the important data of the plant. Wärtsilä uses an own environment for reporting, the Wärtsilä Information System Environment (WISE). The WISE can provide different kinds of reports, logs and plant documentation. (Wärtsilä ELWIS (Electrical Wisdom) 2013.)

The engines are connected to the generators in the generating sets. The generators have AVR-control which regulates the voltage by controlling the excitation current of the generator. The AVR keeps the voltage constant, reduces transients, keeps generators on stable operation range and controls reactive load sharing. (Wärtsilä ELWIS (Electrical Wisdom) 2013.)

Wärtsilä power plants usually have a distributed control system (DCS). This means that different controllers with different processors are connected and operate together. For example the engines can have their own controller (UNIC) and auxiliary equipment of their own. David Lindsley (2000: 153) presents a typical power plant distributed control system. The DCS presented by Lindsley shows central control cabinets which are connected to a data highway. This highway can for example be realized with an Ethernet connection. In addition to the control cabinets, the operator workstations and engineering workstations are typically connected to this data highway. The control cabinets or cubicles contain I/O (Input/Output) cards, CPU, memory and power supply. The hardwired signals which do not go through the data bus are connected to the cubicle I/Os. (Lindsley 2000: 153.)

In Wärtsilä power plants, there are often several PLCs which have own control cabinets located in different places at the plant. This is because of the need to reduce cabling. There are often so many signals and a need for own PLCs for different systems, that it is better to use several PLCs or remote I/Os which are located in different cabinets at different locations at the plant. This makes the distributed control system even larger and increases the requirements in the design of cabling and signal connections.

The I/O cards of the control cabinets have both analogue and digital channels. The signals which are inputs and outputs for the cabinets are the signals in the MoPo-list. The grouping of the signal I/Os in the cabinets is one thing to consider in the cabinet and cubicle design. For example analogue out signals require current generators which take a lot of space. The physical location of the modules and equipment at the plant also affects cabinet design. Lindsley (2000: 156) mentions that the signals to the same system are not necessarily carried in the same cable. Like Lindsley (2000: 156) also mentions, identification of signals is important right in the beginning of the design process.

The signal database can help in this problem. Information, of which parts the signals are connected to, already exists and could be used to determine the location where the signal will be cabled. The locations depend on the plant design but in most cases, a standardized design is followed. One possibility would be to use this standardized design to create a preliminary suggestion of cabinet cabling based on physical location of the parts to which the signals are connected.

2.1.4 Remote connections

Remote connections are used for communication with third parties, for example the communication between WOIS and the grid operator. There are different ways how the remote connections are realized in Wärtsilä power plants. Today, all the communication between the main components is done through Ethernet. The main advantage of Ethernet is that most devices today support Ethernet-communication using TCP/IP (Transmission Control Protocol/Internet Protocol) protocols.

For communication with the customer's Remote Terminal Units (RTUs), Wärtsilä has designed free contacts for hardwired signals in the common control panels. This is a cost efficient way to enable digital and analogue signal communication. To the customer PLCs, RTUs and Distributed Control Systems (DCS), the communication is handled by an RS-485 serial connection. This is a medium cost solution for data exchange on short distances.

The connection to the customer's or network operator's SCADA is normally handled by Ethernet through dedicated firewalls. In addition to Ethernet, a Modbus connection can be used to connect Wärtsilä's control system to customer supplied control.

2.2 Product lifecycle management

One of the reasons to introduce a signal database system is to be able to better maintain and track signal data. This is a part of Wärtsilä's Product Data Management (PDM) programme. A wider term for this is Product Lifecycle Management (PLM) which includes more than just product data. PLM includes lifecycle as a new dimension in product data analysis. One definition of PLM is: "*PLM is defined as a systematic concept for the integrated management of all product related information and processes through the entire lifecycle, from the initial idea to end-of-life*" (Schuh, Rozenfeld, Assmus & Zancul 2008). Basically this means virtual items for all products and designs. These items can be handled in different information systems and data like drawings or cost information can be added to the items. In this way PLM can reduce the need of handling separate documents and helps to connect all related information to the design items.

Saaksvuori & Immonen (2002: 3) define PLM as "*a systematic, controlled concept for managing and developing products and product related information*". The signals in the signal database belong to product related information which is why they have to somehow be connected with PLM. Saaksvuori et al. (2002: 57) present two ways of connecting a database to PLM systems. These are data transfer and database integration. Data transfer means copying information from one place to another, for example to another database. Database integration means using the same database for different applications. These applications can then access the database through an application programming interface (API). Database integration is more expensive and more difficult. However, it provides better speed and the data can be managed in one place. (Saaksvuori et al. 2002: 57.)

When thinking of the PLM advantages and data maintenance requirements, it is clear that PLM systems need to be integrated with the signal database. One of the reasons for PLM implementation is to reduce the need of managing data in several different places. This creates a requirement that PLM software need an API that supports database integration. In the case of the signals, the signal database can be connected to a PLM system where the portfolio of parts and units is managed.

The idea to filter the signal list with the help of quotation software data came from the PLM systems. When creating project structures in PLM tools, the XML (Extensible Markup Language) -file which is an output from the quotation software, is compared with an over-complete list of items. Based on this comparison, a project specific structure of items can be created. This same method can also be used when creating a project specific signal list. In this thesis, the word *over-complete* is used to describe the complete list of all possible signals and parts, not only those included in a specific project.

The development of product lifecycle management started from the need to create product data models for handling data with different computer systems. According to Pierre Hadaya and Philippe Marchildon (2012), PLM systems origin from the combination of CAD (Computer Aided Design) systems and product data systems. Some of the main benefits and reasons for implementing PLM are according to Hadaya et al. (2012) better supply chain collaboration, reducing design time and improving success rate and innovativeness of new products.

In practice, PLM can increase work efficiency by collecting all data to one place and enabling re-usage of data. The need of searching data in several different places is reduced. This is, according to the interviews performed for this work, one of the main problems of the current signal list creation. Re-usage of data includes both using data created in other systems and using data from earlier projects. In the case of the plant signals, the project specific signal list can be generated when the project scope is known. This scope is created in project sales and quotation systems before the actual design work of the project starts. When the scope is known, the parts included in the project are known and therefore it is possible to list the signals which are connected to those parts. This prevents the need of configuring the signal list manually based on the project contract.

Since many of the units in the signal database will be auxiliary modules, the module data management systems should have a connection to the signal database. The modules are for example pumping modules, fuel separators or exhaust gas modules. In PLM, the modules will be managed in a specific module master directory. This directory can give input to the signal database about which modules are in use.

2.3 Signal connections in power plants

In this section the current signal management methods and the data included in the project quotations are presented.

2.3.1 Standard signal configuration list

The current way of configuring the signal list in Wärtsilä power plants design, is a spreadsheet-list where all signals and their data are listed. By using macros in spreadsheets, the user can select the plant configuration and then get a filtered list with the parts and signals needed for that particular plant configuration. This spreadsheet-based signal configuration tool is called the Monitoring Point-list or MoPo-list.

The MoPo-list consists of parts included in power plants and their signal data. The following figure shows an example of a part and its signal data. The header, painted gray, is the part name. The rows below are different signals connected to the part, which in this case is a heavy fuel oil separator. The signal information visible in the picture below is system code, system number, component code and tag code extensions. One can also see that for example the *separator running*-signal is a digital input signal and that it is normally closed.

HFO separator 1						
Separator running	PBB	901	M001	RNG	DI	closed
Separator feeder pump running	PBB	901	M002	RNG	DI	closed
External emergency stop	BJJ	901	S020	ESG		
Separator panel common alarm	BJJ	901	S035	AG	DI	opened

Figure 3. Signals for a heavy fuel oil separator.

The spreadsheet-based MoPo-list has built-in macros which allow the project engineer to make the plant configuration. After selecting the correct engine models and other options, the macro-code filters the required parts and modules needed for that specific plant. In this way the engineer can get a list of signals and information of to which parts they belong. This list of signals is then provided to the manufacturer of PLCs and control panels.

The problem with the MoPo-list is that it may sometimes contain errors. These errors occur because of two main reasons. Firstly, the list configuration rules might be based on old design data. While the power plant technology has developed, new requirements and signals might have had to be added. This has resulted in that engineers just manually add the required data after running the signal configuration.

The second problem is related to the first one. Being spreadsheet-based, the list of thousands of signals is difficult to maintain and there have not been enough resources to update the list every time some changes to the signal set-up happen.

Plant Code	System	System code
	Number	Compoent number
	Comp.	Component code
	Extensions	Tag code extensions
Instrument information	I / O Type	I/O type to PLC (DI, DO, AI, AO, TC)
	Signal	Signal type to PLC (The status refers to action in the name column)
	Range	Range of the signal (e.g. 0-10, 0-600)
	Code	Mechanical code of the instrument
Function / Setpoint	Start	Setpoint for starting the process
Control	Stop	Setpoint for stopping the process
	Startblock	Setpoint for inhibiting the starting of the process
	Load red. [+10s]	Setpoint for engaging the load reduction sequence
Alarm	High	High level setpoint for activating the alarm
	Low	Low level setpoint for activating the alarm
	Delay [s]	Time delay before activating the alarm (sec.)
	Deviation	Setpoint for alarm from deviation of the average
	Delay [s]	Time delay before activating alarm from deviation of the average (sec.)
	Differ. [+10s]	Setpoint for differ. alarm between 2 valvewise
Shutdown	High	High level setpoint for activating shutdown
	Low	Low level setpoint for activating shutdown
	Average	Setpoint for auto stop from the average
	Delay [s]	Time delay before activating shutdown (sec.)
Auto stop	Auto stop	Setpoint for auto stop (e.g. dev. of the aver.)
	Delay [s]	Time delay before activating auto stop (sec.)
	Unit	Unit of measure for the monitoring point
WOIS / WOT	Status	The process operating status is displayed on WOIS screen ("X"=yes; " " = no)
		The monitoring point is included in the common alarm of panel ("CA"=yes; " " = no)
	Trend	The measured value is recorded to historical trend ("X"=yes; " "=no)
	Event	Changes in the operating status are logged to the event file ("X"=yes; " "=no)
Connect. to Panel		The monitoring point is connected to control panel
Connect. to PLC		The monitoring point is connected to PLC system in control panel
Rev.		Revision letter of the monitoring point (a, b, c...) for project specific changes
Note		Project specific note or comment

Figure 4. Signal data with explanations in the MoPo-list.

Figure 4 shows all the attributes which the signals can have. The first column is the group of attributes. The second column shows the shortened attribute name and the third column the complete explanation. The “Function/Set Point”-group is still divided into four different functionalities. All signals do not have every attribute shown in the list. The attribute list contains some abbreviations that need to be explained. The WOIS field means Wärtsilä Operator’s Interface System. This is the operator’s interface for the plant control which is usually located in the main control room in the power plant area. The signal attributes in Figure 4 vary depending on the signal. For example only analogue signals have a *Trend*-value.

2.3.2 Metadata in project scope files

The following figure shows how the metadata is structured in XML-files.

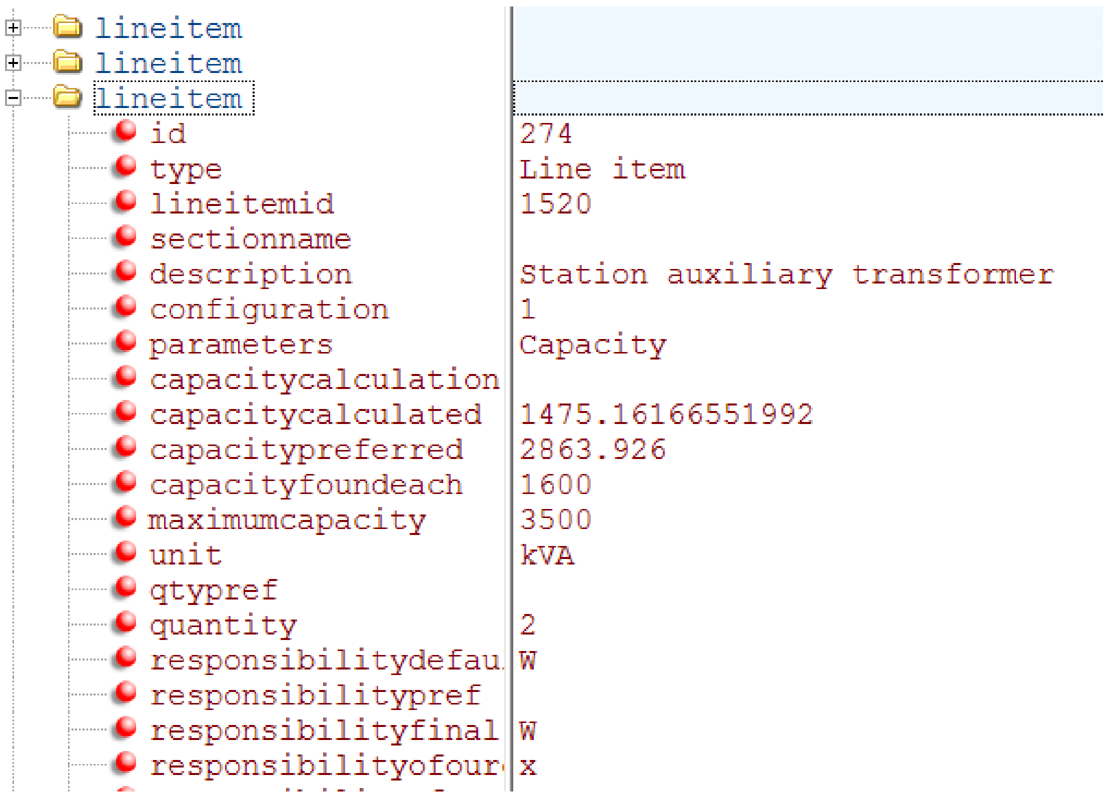

Figure 5. Example of an item in the quotation XML-file.

One method to get the project specific signal list is a comparison between the XML-file, which originates from the sales tool, and the over-complete signal database. The sales tool XML-file is generated when it is time to provide the project quotation for the customer. The main purpose of the sales tool is to make the project configuration. The outputs are quotation documents, scope of supply and technical specifications. The sales tool generates an XML-file which contains the scope of supply, cost data and technical data (Sales Tool Introduction 2013).

The example of Figure 5 is from the sales tool XML-file. Because the XML-file already contains a lot of information about the project scope, it is unnecessary to make the configuration again in the MoPo-list. All the parts of the power plant have a unique ID, called “lineitemid”. This ID is already used when creating project structures for Wärtsilä’s PDM information systems. In this way it is possible to find the items included in the project scope from the over-complete list of items. This same method can be used for the parts in the signal list as well.

At the moment a new sales tool is taken into use and the example of Figure 5 will not apply anymore. However, XML is a very common output type and it is possible to get XML output from the new sales tool as well. When designing the signal database solution, one thing to keep in mind is that it should have support for all the common input types. This is to ensure that comparison with the project quotation could be possible.

The comparison of the XML-file and the over-complete signal list, requires an ID which can be used for comparison. At the moment however, there is no unique ID for the parts in the MoPo-list. The plan is to make a running number ID which would be a unique ID for the signals. This ID will have to be generated when the signal is added to the signal database and it would tell to which product portfolio part the signal has a connection. The mapping of the signal IDs to the sales tool IDs has to be stored for example in the signal database to enable the comparison. If the signal database will be connected to the master data product portfolio, this ID would be easy to copy to the signal database.

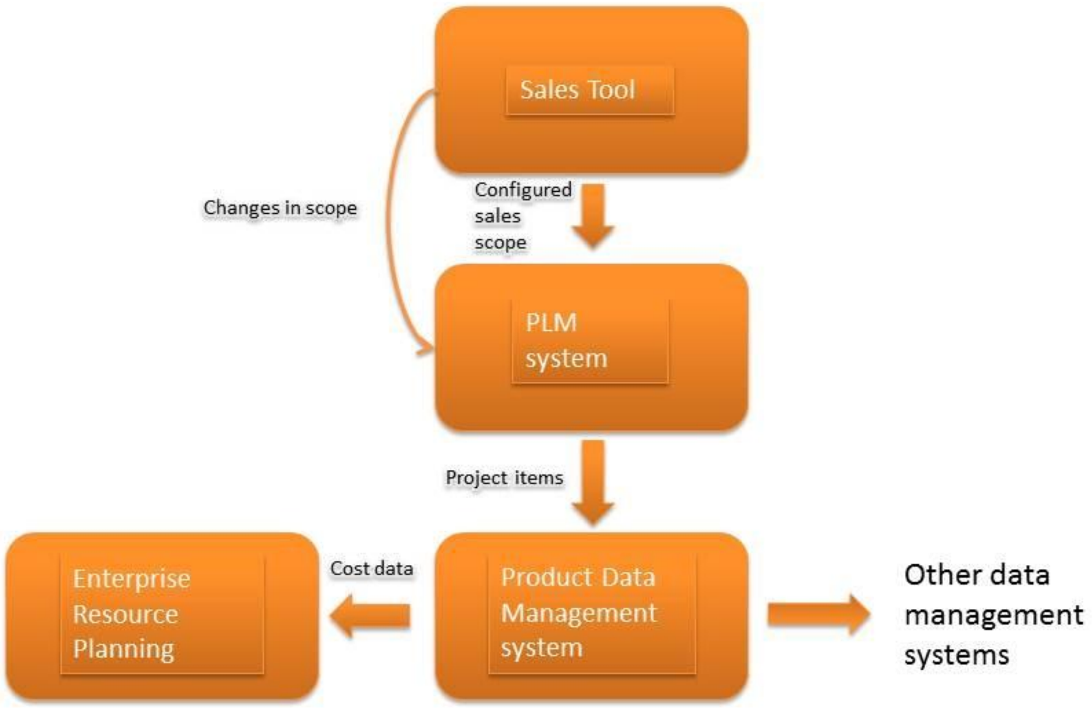

Figure 6. Data flow in PLM project data import.

For the generation of the project specific signal list, the best option would probably be to use the configuration management PLM tool from which it is easier to get an approved scope file. This will be discussed more later in this thesis. The XML output generated by the sales tool is read into the PLM tool and managed there. For potential changes in scope, the PLM tool is a better place to pull the project scope file from.

As an example of how the project scope items can be imported to data management systems, the current PLM and PDM data import can be used. Figure 6 shows this data flow. In this system, the project data is pulled into the PLM system from the sales tool via a data processing server. The data can then be sent further to PDM and ERP (Enterprise Resource Planning) systems.

3 DATABASE THEORY

A database contains information that can be used by different applications. Colin Ritchie (2008: 10) mentions this differentiation between data and applications as one of the main characteristics of a database and that it enables the same data to be used by different applications. According to Ritchie (2008: 10), a database can be defined as something that “holds data as an integrated system of records” and “contains self-describing information”.

The data in the database is described by attributes and there are usually relations between different data types. A common way to model the data is to use classification. Data is classified into different classes and these classes have attributes which describe the data. At least one of the attributes is a unique ID which can be used to identify specific data objects. These types of models are referred to as relational database models.

A commonly used method to describe the relations between different database entities is the ER-diagram. This is a simple way to model data objects and their attributes and can be converted to the already mentioned relational database models. These both models are conceptual models and do not represent real database solutions. However, before starting to create a real solution, a conceptual model needs to be done.

3.1 Development model

When developing new IT solutions, a development model is normally used. Some common methods are structured design, Rapid Application Development (RAD) and agile development (Tegarden, Dennis & Wixom 2013: 6−18). These categories have several different models under them.

The structured design models, like the waterfall model, have been used a lot in the past and are still popular in many development projects. During the last decades the agile models have increased in popularity. The best advantage of agile models is shorter development times. Many of these models contain parallel development and the project is divided into sub-parts.

Tegarden et al. (2013: 9) divide the system development lifecycle of structured design into the following parts:

- Planning
- Analysis
- Design
- Implementation

When thinking of the scope of this thesis, the phases of planning and analysis will be covered. The main part of the planning phase is identifying why a new system needs to be built. This includes how much benefit it will bring for the organization and if it is technically and economically possible. Things like database solutions and project costs need to be thought about already in this phase. Planning also includes project management planning. The schedule, budget, human resources and project steps need to be defined. (Tegarden et al. 2013: 4.)

The analysis part contains analysis strategy, requirements gathering and developing a system concept. A simple way to make the analysis strategy is to describe the current system (As-is) and the wanted system (To-be). This method is commonly used in information technology development projects. The requirements gathering will be discussed more in detail later. Basically it contains gathering the information from future users of the functionalities and required outputs of the information system that will be developed. The system concept is a high-level model of the system architecture and its functionalities. The concept is often used for making the decision to continue the project after the analysis part is ready. (Tegarden et al. 2013: 4−5.)

After the planning and analysis phases are ready, the design of the actual system starts. The design phase contains design strategy, basic architecture, database specifications and programming (Tegarden et al. 2013: 5). The final phase is implementation where the system is constructed and tested. The installation and take-to-use of the system are not the final phases, support and training is important to ensure successful implementation. When the implementation is done, the maintenance and continuous support starts. This goes on until the system is someday taken out of use.

Tegarden et al. (2013: 5) mention two common models which belong to the structured design category. These are the waterfall and the parallel development model. The waterfall model (Figure 7) is best suitable for environments where the requirements are well understood (Guntamukkala, Wen & Tarn 2006). The planning and analysis phases of the signal database in this thesis can be considered to be done in this kind of an environment.

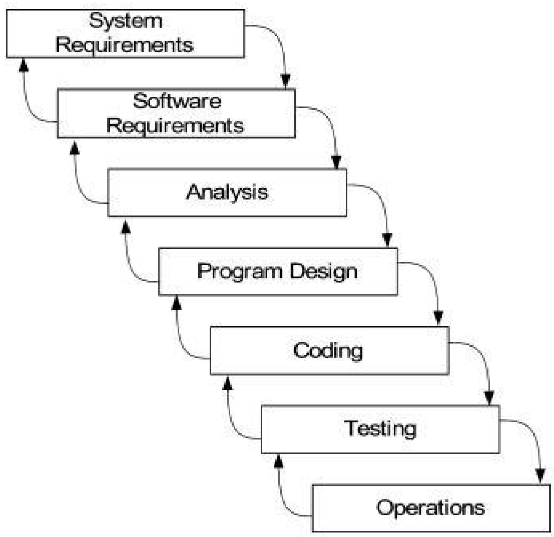

Figure 7. The waterfall model (Guntamukkala et al. 2006).

Because this thesis is written by only one person, the waterfall model is a suitable alternative for the planning and analysis phases. Many of the other development models include splitting the development for different teams, for example in the parallel development model. The description of the analysis phase by Tegarden et al. (2013: 5) matches quite well with the plan of this thesis. Therefore the waterfall model is something that can be used in this phase of the signal database design. Most of the other development models contain the same steps but they are just done for smaller parts of the information system.

Ian Sommerville (2011: 50‒52) presents the Rational Unified Process (RUP) system development model. This is a modern process model which is closely related to UML modelling. The development model supports iteration and is suitable for most IT system development projects. The workflow of the RUP consists of *business modelling*, *requirements*, *analysis and design*, *implementation*, *testing*, *deployment*, *configuration and change management*, *project management* and *environment*. (Sommerville 2011: 50‒52.)

The first three RUP workflow steps are of interest in this thesis work. The process starts by modelling the use cases and identifying which business processes are involved in the system which is developed. The second step is to identify the requirements and stakeholders. After this, the architectural design can start in the third step. The next section (3.2) will explain UML, which can be used for this. Section 3.3 will discuss requirements analysis. Sommerville & Sawyer (1997: 164) explain that the main benefits from system modelling (first step in the RUP) is that it reveals inconsistencies and helps in the analysis of requirements.

3.2 UML

UML is a widely used and well adopted language in ER-modelling. The use of UML has increased when object oriented thinking has increased in database design (Naiburg & Maksimchuk 2001: 11). UML is programming language independent which enables it to be used for almost any programming types (IBM Technical Library 2014). UML can be divided into the following diagrams (IBM Technical Library 2014):

- Use-case diagram
- Class diagram
- Sequence diagram
- State-chart diagram
- Activity diagram
- Component diagram
- Deployment diagram

These are different diagrams that can be used to describe data content, workflows, roles, responsibilities and relationships. For the usability of the database it is good to make use-case diagrams which describe the activities that should be possible to perform in the database solution.

Tegarden et al. (2013: 28−29) mention Object-Oriented Systems Analysis and Design (OOSAD) as the modern way to do system development and it can better take into notice both the processes and the data. OOSAD is use-case driven, architecture centric and iterative and incremental (Tegarden et al. 2013: 28−29). Use cases are often the first way to start to describe a system. With the help of the use cases, the requirements can be gathered. The benefits of use cases are that they are simple and focus on one function at a time. This makes it easy to see what the system must be able to do.

Tegarden et al. (2013: 29) say that in OOSAD the model usually includes functional, structural and behavioral architecture views. The iterative and incremental nature means that the developed system model is continuously improved and several evaluations and improvement rounds are made. The system model usually grows during the iteration rounds. Tegarden et al. (2013: 29) say that the biggest benefit of the OOSAD development method is the possibility to have modular development. The system can easily be broken into modules which are also easy to build back together (Tegarden et al. 2013:

29). The OOSAD method is often used combined with UML. With UML, the use cases and architectural views can be described.

The use of different UML processes is described in the *The Unified Process* methodology. This methodology helps to tell when different UML models should be used and what phases there are in an information system development project. The different phases of *The Unified Process* are inception, elaboration, construction and transition. In OOSAD development, this process methodology is commonly used.

Usually the use case model is one of the first models to start with in information system development. With the help of use cases it is possible to find the main requirements and functionalities that are needed. Use cases are a part of the functional requirements specification. Activity diagrams, which resemble swim lane diagrams, can be used to show workflows with the roles of actors included. A sequence diagram is almost like an activity diagram. The difference is that a sequence diagram concentrates more on data flows while an activity diagram shows tasks for different stakeholders. These three UML models are suitable for representing the first part of the system development where the main functional requirements are gathered. The other UML models are more focused on structural views, which are not essential in the first phase of the development.

Naiburg et al. (2001: 18) start the business modelling design by drawing use cases, activity diagrams and sequence diagrams. These models can be used for the functional requirements gathering. After this, more detailed models about data and its relations can be drawn.

3.3 Requirements management

A requirement is "a need that must be satisfied by the creation of a physical product or system" (Brace & Cheutet 2012). These needs will come from the people or organizations that will use the designed product. The needs can be functional, quality related, security related, performance related or connected to maintainability and lifecycle management. The requirements define the functionalities and properties of the developed system or product. Conger (2012: 47) defines database requirements as “something the database must do in order to meet the business needs of an organization”.

Before starting to create a model of the database, the requirements have to be gathered. In the signal database system, there are three main users: the R&D department who maintain the signal information, the project department who create the project specific signal lists and the suppliers of the control systems who use the signal list when designing the control cabinets.

Murali Chemuturi (2013: 5) defines the requirements management process as:

1. Documenting requirements.
2. Analyzing the requirements.
3. Tracing the requirements through out the development lifecycle.
4. Prioritizing the requirements, especially their order of implementation.
5. Agreeing upon the requirements, that is requirements are accepted for implementation by the project team and are approved by the stakeholders.
6. Controlling the requirements, that is controlling the change to the agreed upon requirements.
7. Communicating the status and progress of implementation of requirements and changes received thereon to all stakeholders.

In the case of the signal database, interest is especially put on documentation of the requirements, analyzing them for the database model and prioritizing the most important needs.

Chemuturi (2013: 13) introduces three ways to classify requirements. These are according to functionality, according to product considerations or according to source considerations (Chemuturi 2013: 13). The method used in this thesis is mostly source consideration. This means gathering requirements from the users who act as sources for requirements. Because of the simple nature of the signal database, the construction or architecture of the software will most likely not be an issue. The most important thing in the construction point of view is the database model itself which includes classes and relations of the database objects.

The functional requirements are partly integrated with the source requirements. Some of the requirements brought up in the interviews were good usability and requirements of what should be possible to do in different stages of the project work. These can be classified as both functional and source requirements.

Naiburg et al. (2001: 54) define the goals of requirements gathering as:

- Establish the scope of the planned system.
- Established a detailed understanding of the desired capabilities of the system.

Databases often have relations to other databases from where they get information. In addition, the database needs some kind of application user interface for accessing and using the data. These requirements might cause the scope to grow unnecessary large and one have to keep in mind that the database itself is what needs focus in the beginning. Since the signal database will contain technical information of Wärtsilä Power Plants, security is an important issue. This can include access rights, data encryption and physical location of data. For example, if the database should be on premise or in the cloud.

Conger (2012: 47) divides database requirements into the following categories:

- Data requirements
- Report requirements
- Access and security requirements

Looking at this categorization, one can see that the role of reporting and security is just as important as the data itself. Without proper reporting functions it can be difficult to use the data efficiently. Data security is always something that has to be considered already in the beginning of the development of an information system. If the security is left to later parts of the development, it can be too late to fix security issues.

When gathering requirements, the answers from different stakeholders can be quite different because everybody looks at the information system from their own point of view. This might make it difficult to analyze the requirements and use them for the system development. One way to analyze requirements is to use check-lists as Brace et al. (2012) have done. According to Brace et al. (2012), the interest in using pre-defined models in requirements engineering has grown. Reasons to this are different inputs from different stakeholders and the variability in terminology used. For design engineers, pre-defined models can make it easier to collect the essential requirements right from the beginning. The end-user might not think about all the things that are crucial for the designers to know in an early phase. (Brace et al. 2012.)

As a part of the check-list requirements gathering, a technique which helps to categorize and manage requirements can be used. This technique includes the following four phases (Brace et al. 2012):

- Abstraction
- Decomposition
- Projection
- Modularity

Abstraction is to ignore details and concentrate on the main things that the requirement needs. Decomposition is to break large-scale requirements into parts because sometimes a single requirement can require several properties of a product or system. Projection is related to viewing the requirements from a certain point of view to understand them more clearly. Modularity means to find requirement structures that are stable over time and can be re-used. As one of the greatest advantages of this kind of a check-list structure, Brace et al. (2012) mention easier integration to PDM and ERP systems. Another advantage is to more easily find the responsible person for creating a solution to a particular requirement. In the design process, using a check-list based approach can improve the design quality through better formality which reduces the risk to miss important requirements. (Brace et al. 2012.)

One approach to gather requirements can be to formalize them after discussions with different stakeholders. Check-lists might not be needed before starting the discussions. Sometimes it can even be better to avoid too rigid requirement structures in the beginning of the requirements analysis. This is because the requirements might change and there might come a lot of new requirements during the development.

Sommerville et al. (1997: 63−111) point out that it is sometimes not enough to just ask for requirements, the organization should also be analyzed. This is to identify different roles and different understandings of the planned system. For example, people with no development responsibility might not think about the costs of implementation. The requirements are also often dynamic, especially in changing economic situations new requirements might emerge. Sommerville et al. (1997: 63−111) also advice to keep the business concerns in mind. The question of what business benefits the new system will bring needs to be answered. These benefits can for example be better quality or faster delivery times.

Paying attention to all different stakeholders and getting different viewpoints for the requirements are things which Sommerville et al. (1997: 63−111) mention several times. This is important for avoiding missing requirements. Using stakeholder inputs can also help to structure the requirements in practical groups. One beneficial aspect can be that if the stakeholders are involved in the development, they accept the new system more easily. Using scenarios and defining operational workflows are good tools for the requirements analysis. This is something that UML can be useful for. (Sommerville et al. 1997: 63−111.)

4 DATABASE REQUIREMENTS

The main research method of this work is to perform interviews with the main stakeholders in the power plant signal configuration process. These stakeholders are the R&D department, project management and the control system suppliers. The following sections present summaries of the interviews. The main goal of the interviews was to find out current problems, requirements and improvement suggestions related to the signal configuration process.

4.1 Data and maintenance requirements

The R&D department will be the ones who create and maintain the signal data. A meeting was held 23.1.2014 with the electrical & automation department (R&D) to discuss the database development. In this meeting some of the main requirements considering the data and maintenance of the signal database were discussed. (Wärtsilä Power Plants Electrical & Automation-team meeting 2014.)

The coding and naming of the signals were discussed. It is a lot of work to change the coding system and a big question is which standards should be followed. One good point about following an international standard instead of Wärtsilä coding, would be that an international standard can more easily be translated into a new coding system if needed. The naming of the signals has been a problem to the automation system suppliers. The signal database would need a standardized way of naming the signals. A naming example could be to use some of the signal data in the signal name. For example action type, function etc. In addition to short and long signal name fields, a third field with instructions could be added. (Wärtsilä Power Plants Electrical & Automation-team meeting 2014.)

Alarm prioritization is one thing that would be good to add as a signal property. As an example, a connection fault alarm should override other alarms because the signal connections cannot be trusted. This can be done by defining alarm priority groups and adding them as an attribute to each signal. (Wärtsilä Power Plants Electrical & Automation-team meeting 2014.)

The new signal database would be the main database of all signal information. Based on the signal database, the PLC and WOIS design would be made. This means that there has to be enough information about the signals to efficiently use the database as a base for designs. There was already some discussion in this thesis about the object-oriented structure of the planned database. Many parts or units can have the same signal so the database needs to be designed in a way that the parts or units know what signals belong to them. The signals do not necessarily need to know what part or unit they belong to. This helps to make the maintenance easier because the signal information needs to be maintained only in one place. (Wärtsilä Power Plants Electrical & Automation-team meeting 2014.)

4.2 Project side requirements

The main users of the signal configuration tool are the project engineers (PEs) who create the project specific signal lists while working with different projects. At the moment, they do the project signal configuration by using spreadsheet software macros. In their work, the PEs do the signal configuration by gathering information from the project contract and the detailed design and comparing it with the macro-generated signal configuration. Often changes are required and the PEs might have to manually search information from different standard signal lists.

In an interview held with one of Wärtsilä Power Plants' Senior Project Engineers, the following things were mentioned as the two biggest problems today when using spreadsheets for creating the project specific signal list (Senior Project Engineer, Electrical, Wärtsilä Power Plants 2013):

1. Data in the spreadsheet-file is not up to date.
2. Errors in the part coding, the code might need to be manually checked which creates a lot of unnecessary work.

Both of these problems are related to difficult maintenance of data which is spread on several spreadsheets. Because of the large size of the power plants, the amount of signals is large and it can be a lot of work to search and update a specific signal value. Another problem with spreadsheets is the distribution and update notifications. PEs might have old versions of the signal configuration tool saved on their computers.

Another aspect of signal coding is finding an unused code when adding a new signal. At the moment, the user needs to search for a code which is not in use yet by entering random codes in spreadsheet search functions. The coding for new signals should be automated to reduce work and to avoid errors like duplicate codes. (Signal database meeting, Wärtsilä Ship Power & Power Plants 2013.)

Today Wärtsilä is using an online internal document management system for data sharing. So far the MoPo-list has been distributed out by using this document management system. This system however, starts to be old-fashioned and is not so flexible for handling large project structures. It is however already possible to send out messages to users when the MoPo-list is updated. Of course in this case all the users need to be known. With the proposed database for signals, the newest version would always be available. The old versions could be hidden and made available only if the user specifically asks for older versions.

Another discussed topic in the interview of the PEs was user-friendliness of the proposed database system. For example how the output of the new system would look like. The proposed plan is to include a graphical user interface where the user could see the signals generated based on the project quotation file. In addition to the graphical user interface output, it was discussed that a spreadsheet-file would be a useful output type. This is because of the frequent need of changing things in the signal list. Although the proposed database would improve data quality, there might always be errors which need to be corrected. In a spreadsheet-file it would be easy to make the corrections. Of course it should also be possible to make the corrections in the graphical user interface. (Senior Project Engineer, Electrical, Wärtsilä Power Plants 2013.)

Some parts or systems have options for optional signals. The addition of these signals depends on the project and which systems the part will be connected to. In the user interface of the database system, there should be the possibility to select these optional signals. They should also be flagged so the user can clearly notice which signals are optional.

One thing that was proposed in the meeting was the possibility of adding signal ranges automatically based on the project quotation file which contains a lot of information of for example ambient conditions. Another idea was to interview the end-users of the signal lists. The end-users manufacture the control cabinets where the signals will be connected. The possibility of determining the signal ranges based on quotation file content will be kept in mind but it is a bit out of the scope of this thesis. After the development of a functioning database solution, the automatic range settings should be investigated further. (Senior Project Engineer, Electrical, Wärtsilä Power Plants 2013.)

A second interview was carried out 26.11.2013 with PEs from the project side. One person interviewed was a senior chief project engineer and the other a project engineer, both from the electrical side. Some new aspects that had not been talked about in the first interview were found. Concerning the database system and the creation of the project specific signal configuration, one thing that needs to be remembered is that if Wärtsilä has only a small scope in a power plant project, the PEs do not according to the interview, get a quotation file from the sales tool in some cases. This quotation file does however exist and it could be used for generating project specific signals. Despite the small scope, there is still often a need to get the complete signal list for the project. This requires that it should be possible to make the project configuration manually in the new signal database system. It was proposed that this configuration would be similar to the one that already exists in the spreadsheet-based MoPo-list. This would make it easy for the PEs to adapt to the new database system. (Senior Chief Project Engineer, Electrical, Wärtsilä Power Plants & Project Engineer, Electrical, Wärtsilä Power Plants 2013.)

As in the first interview, this second interview also showed that outdated information is the biggest problem of the current spreadsheet-based MoPo-list. It was brought up that PEs cannot always trust the data. Especially larger modules with many signals were mentioned as a problem. The PEs often have to contact people from the R&D department for confirmation that the information is up to date. (Senior Chief Project Engineer, Electrical, Wärtsilä Power Plants & Project Engineer, Electrical, Wärtsilä Power Plants 2013.)

In a training session held 7.3.2014 called *MoPo-list and process drawing balance*, some other issues concerning the current MoPo-list came up. Many of these were related to engine signals which are provided by Wärtsilä Ship Power. Some of the engine instrumentation information is missing and it is difficult to know where to get the latest version of the engine signal list. These issues show that it is worth to consider integration between the engine signal database and the power plant signal database. (MoPo-list and Process Drawing Balancing 2014.)

Some problems which are caused by limitations of the current MoPo-list were also brought up. For example in fuel tanks, there are often two high level indicators. The first one acts as a monitoring sensor which gives a signal when the tank is full and stops the feeder pumps for the tank or controls a valve to steer the fuel flow through another route. This is normal automation and should not cause an alarm. However, the second high signal, usually called *high high*, acts as a backup signal and gives alarm to the power plant operator’s interface when the fuel level reaches the *high high* sensor. At the moment however, there is only one spreadsheet-cell for the high alarm so the basic *high* indication for normal automation is missing from the MoPo-list. (MoPo-list and Process Drawing Balancing 2014.)

A second issue was related to the generation of the I/O calculation. The I/O calculation is a feature of the current signal configuration tool and it counts and groups the different signals. These are for example analogue, digital, pulse or temperature sensor signals. The I/O calculation is used by the control system suppliers to determine how many cards are needed in the control cabinets and the calculation gives them a specification that is used to calculate offers. If the attribute *Connect to PLC* is empty, the signal will not be included in the I/O calculation. At the moment there is no better way to generate the I/O calculation. If the project engineer happens to forget to fill this value or does not know if the signal should be connected to PLC or not, the control cabinet manufacturer might manufacture cabinets which do not have enough signal connection places. This results in that connector cards have to be added later which is unnecessary work. (MoPo-list and Process Drawing Balancing 2014.)

Data problems that were brought up include data of newest modules missing, problems with the *range* value, and filling in the *unit* and *trend* fields. The range of the analogue signals vary a lot so it is difficult to put any default value in the MoPo-list. Both the signal ranges and project specific lists are things that could be handled with a database solution. As mentioned earlier, the complete list of signals and parts can be compared with the project scope from the sales tool to create a project specific signal list. The *unit* value can be a bit unclear. It tells the unit of the range, which is the unit to be monitored. The *trend* field should only be filled in for analogue signals. This could be marked more clearly. One solution could be to prevent filling in values which do not relate to the signal. (MoPo-list and Process Drawing Balancing 2014.)

An interesting topic brought up in the training session was how the As-built signal list should be followed and if there even are possibilities to get a signal list of the As-built situation. The signal list always changes during the project and it is first at the power plant site when the plant is built, where the final As-built list can be done. The current MoPo-list is normally updated until the detailed design phase. When thinking back on PDM, As-built is something that would need to be known for proper maintenance and traceability. However, a question is who would do the updating to get the final As-built list. The signal database would likely give a good platform where the As-built lists can be saved and managed. (MoPo-list and Process Drawing Balancing 2014.)

4.3 End-user requirements

The end-users of the MoPo-list are the companies which provide the automation and control systems for Wärtsilä Power Plants. Interviews with these suppliers have been performed to gather the requirements and needs of the end-users. In this way all the stakeholders can be involved in defining the requirements of the new signal database.

In the first supplier interview, a company which supplies control cabinets and control and automation systems was visited (Supplier 1 2013). The interviewed persons were the engineering manager of automation and control systems and two engineers, one for hardware and one for software design. When receiving the MoPo-list, the supplier starts to design the control cabinets and the automation systems with the following steps:

1. Identifying project items and their based-on models.
2. Performing I/O calculations.
3. Identifying Ethernet connections/locations.
4. Identifying the detailed design information including signal attributes.

In the first step the supplier uses the MoPo-list to identify all parts and modules included in the power plant project. The supplier has an own database for the parts included in the MoPo-list. The supplier normally uses this own database to check which signals and included data will be a part of the project. The second step is to use the I/O-calculations, which are included in the MoPo-list spreadsheet-file, to identify the number of signals for the project. This is one of the first pieces of information needed when starting the design of control cabinets. After this, the supplier needs to know which signals are connected to the Ethernet bus. These are the main usages of the MoPo-list, most of the remaining information needed, is taken from the own database of the supplier. This database is built based on previous projects and communication between the supplier and the R&D department of Wärtsilä Power Plants. (Supplier 1 2013.)

The supplier mentioned that especially the alarm limits for the signals may sometimes be wrong. The signal tags can also be problematic. The component code and the tag extensions included in the MoPo-list are normally used as references when identifying the signals. Without this tag code, the supplier has no way to identify specific signals. However, even the tag codes are not always correct and the supplier mentioned that following some international standard for coding could be useful to help keep the coding correct. (Supplier 1 2013.)

When discussing the attributes of the signals, the supplier mentioned that the (log-) deadband of the signals would be a useful attribute to be added in the signal information. The deadband is the neutral area between upper and lower triggering limits of a signal, this is used for example to avoid triggering back and forth when the signal level is close to the limit. As unnecessary in the MoPo-list, the supplier mentioned information of internal signals in the control cabinets. The supplier already knows what internal signals will be in their cabinets but for Wärtsilä engineers it might still be good to include those in the MoPo-list to get a full picture of all plant signals. (Supplier 1 2013.)

As improvement suggestions or problems, the supplier mentioned the following things (Supplier 1 2013):

- Consideration of including PLC signal blocks used in PLC programming software.
- Signal tags need to be well defined and correct. They are the only identifier of the signals in many cases.
- Naming of parts is a problem when translating to other languages, different names might be used for the same signal.
- Description text size limits, would be good to have one long text and another shortened.

The PLC programming signals are signals in the programming software. For the supplier, it is important to know these when designing and manufacturing the automation systems for Wärtsilä. In large plants with complex systems, the amount of PLC block signals can get really large so it has to be considered if the PLC block signals should belong to the MoPo-list. (Supplier 1 2013.)

The signal coding and tags are issues already brought up in interviews with PEs. Because these tags are often the only way to identify a specific signal, it is very important that they are correct. Wärtsilä has used an own coding system for the parts, although there have been discussion of the need for an international standard. This however, is not in the scope of this thesis so the coding problem can be concluded by saying that the planned signal database should support different coding systems and it should be possible to change the coding system in the future. (Supplier 1 2013.)

The two other improvement suggestions are related to text attribute data of the signals. Because of international projects, the signal names often need to be translated to different languages. The problem is that signal naming might differ in different places and this results in manual extra work when translating signal names. In the planned signal database, one option could be to allow only certain names for signals. These kinds of default values to choose from could be used in other signal attributes as well. This can help to keep data accurate and consistent. The interviewed supplier also mentioned that they have a need for both a shortened and a longer, more comprehensive description of the signal. The attributes in today’s MoPo-list might not be so clear for people who do not use the list so often. Because of this, a longer description would be useful. Many systems also have limits on text fields so one attribute of the signal could be a shortened version, this can then be used in those systems requiring a short text description. In addition to this, the supplier also confirmed that the previously discussed topics of marking customer/Wärtsilä scope and added/removed signals, is useful for them and should be included in the signal database. (Supplier 1 2013.)

The second supplier that was interviewed had quite similar opinions as the first supplier. Data accuracy was told to be a problem with the current MoPo-list. In addition to that, some of the signal attributes were considered to be a bit unclear and difficult to understand sometimes. The work steps of this supplier are the following (Supplier 2 2014):

1. Make I/O calculations for offer.
2. Remote I/O identification.
3. Start of hardware design process.

The first problem considering data accuracy was the “x” mark in start and stop limits. This “x” is there because the project engineers have not specified a value. In some cases the values might be standard but sometimes there are custom signals included so the values would need to be marked. Supplier 2 (2014) mentioned that the range and unit of measurement are some of the most important pieces of information. There are also often confusions about marking where the signals go. For example connections to WOIS and PLCs might not be marked although the signals should be connected to those places. One thing mentioned by the supplier was that the hardwiring method is sometimes unknown. For example if the signal should go through emergency stop or through PLC. (Supplier 2 2014.)

Differences in data between the MoPo-list and the flow chart for the project also cause confusion sometimes. The supplier might not know which one is correct when they show different things. Signal coding also causes some trouble. The combination of the values in the “system”, “number”, “component” and “extension” fields of the MoPo-list, should form a unique identifier which is often the only way for suppliers to identify the signal. However, because there is no duplicate check and there are many changes/additions to the list, sometimes two signals can have the same identifier. In addition, the revision of the MoPo-list provided to the supplier might be outdated. (Supplier 2 2014.)

Based on the interview, the following list of improvement suggestions can be created (Supplier 2 2014):

- The “x” values in the signal attributes should be clearly marked for the project engineers to be filled out.
- Descriptions of the signal attributes should be clearer.

- The signal list has to match with the project flow chart.
- Signal coding has to be unique.
- A column which tells what signals should be connected to power plant external communication would be good to include.
- A “general fault” column could be added because all faults are not high/low-values.
- Signal naming should be consistent, now the same signals can have different names in different places.

4.4 Reporting requirements

In meetings with different stakeholders in Wärtsilä Power Plants, there have been discussions about reporting needs of the proposed database system. Based on what role people have in the business, there are different needs for reporting. For example the R&D department might need reports containing some specific technical values, the project department might need to generate reports for customers and the end-users who manufacture the control cabinets need reports used in assembly.

These reporting requirements need to be considered when designing the user interface and the functions of the signal database system. First of all, users need to have different roles, which will be discussed later in this thesis. Based on what role they are logged in with, there could be possibilities for different reports. This might simplify reporting if there are needs for many different report types.

5 DEVELOPMENT OF SIGNAL DATABASE SPECIFICATIONS

The scope of this thesis is to gather requirements and develop a model that can be used as a base for designing a database for the signals. In this chapter the requirements will be summarized and some basic views of the database solution will be developed. Focus will also be put on use cases to get a clear picture of how the future database would be used. In addition to these, user roles and access rights will be considered because they are important parts of the information system. Figure 8 shows the scope of this thesis. It shows how the data moves through different business phases and how the signal data is located in the business steps. As highlighted in the figure, this thesis focuses on the database solution and the creation of project specific signal lists.

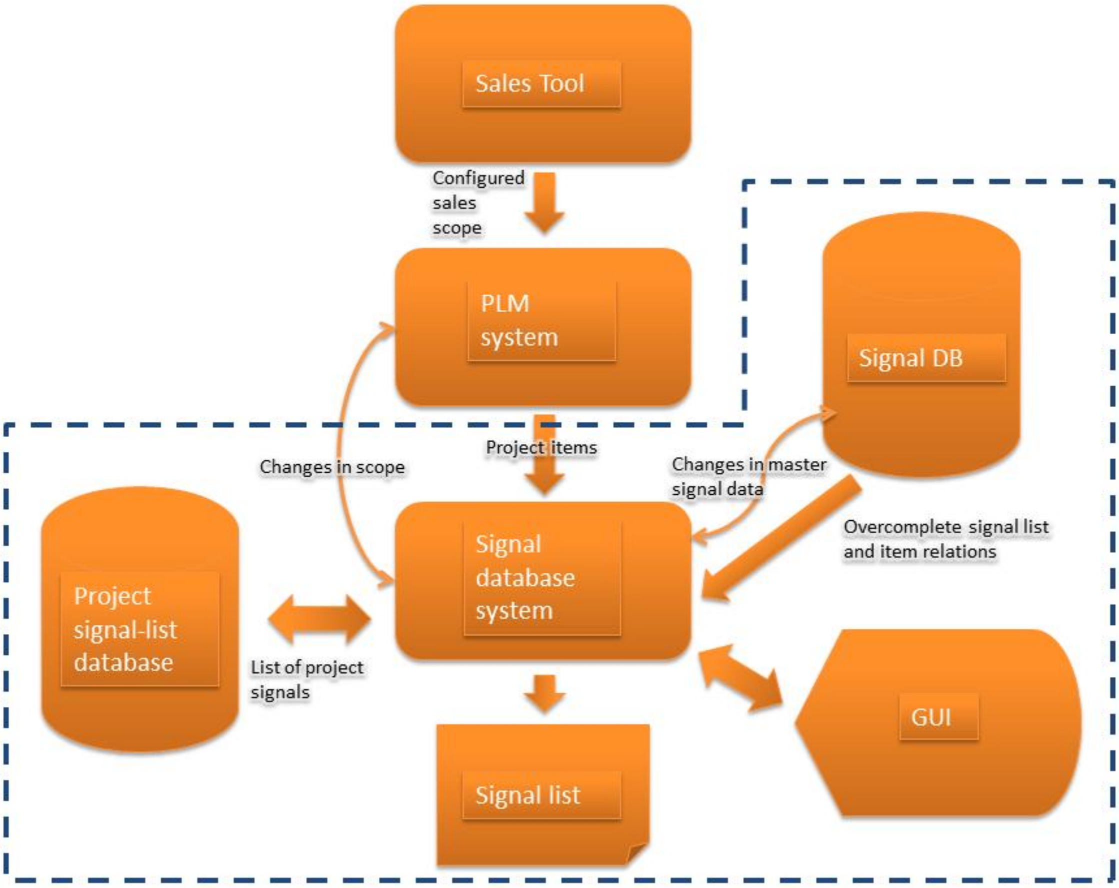

Figure 8. Scope of the thesis.

The signal database of Figure 8 actually contains many lists and databases at the moment. The goal is to find the connections between these so that all power plants signals would be available from one place. The GUI abbreviation stands for Graphical User Interface.

5.1 Main requirements

Table 1. List of requirements.

Role	Requirement
R&D/ Suppliers	International coding standard
R&D/ Suppliers	Standardized signal naming
R&D	Alarm priority groups
R&D	Integration between engine signals and power plant signals
R&D	An attribute fir the *high high* -alarm
R&D	I/O calculation to be more clear (not only check if *Connect to PLC* is filled)
R&D	*Trend* possible to fill only for analogue signals
R&D	Possibility to get an *As-built* signal list
Project engineers / Suppliers	Unique signal coding
Project engineers	Newest MoPo-list version easily available
Project engineers	Spreadsheet-file as output type
Project engineers	Possibility to make corrections/changes in the GUI
Project engineers	Optional signals available and marked
Project engineers	Possibility to configure the complete MoPo-list manually
Suppliers	Unique tag codes
Suppliers	Attribute for log-deadband
Suppliers	PLC-signal blocks included in the MoPo-list
Suppliers	Both short and long signal description text
Suppliers	Unique signal naming
Suppliers	Marking of customer / Wärtsilä scope
Suppliers	“x” values need to be filled by PEs
Suppliers	Clearer attribute descriptions
Suppliers	Signal list to match with the project flow chart
Suppliers	An attribute for connection to the plant external communications
Suppliers	A *General fault* attribute

The requirements found in the interviews that were performed can be grouped by stakeholder role. In this case the main stakeholders can be considered to be Wärtsilä Power Plant's R&D department, project engineers and external suppliers. These represent the people who maintain, create and use the MoPo-list.

Table 1 shows a summary of the requirements found while performing the interviews with different stakeholders. Some requirements have been mentioned by several stakeholders but in this table duplicate requirements are removed to keep it shorter.

The requirements include both data details of the signals and general usage of the signal database. As the main requirement which has been brought up by PEs, one can mention up-to-date data available in a centralized place. As a summary of the interview results in the PE interviews, the difficulties to find signal data from several different places and the quality of the data are things that cause problems today. This in turn results in that the suppliers of the control systems might not get data which has good enough quality. This has also been noticed in supplier interviews. Some suppliers use their own signal databases because it makes it easier for them to generate the data needed for control system design.

If looking at differences between requirements of the R&D department, project engineers and suppliers, one can see that R&D focuses on new attributes and signal coding standards while PEs focus on the usability of the system. Suppliers again, focus on data quality of the signal data.

As mentioned earlier, the main users of the signal database are the PEs who generate the project specific signal lists. The usability is very important to easily be able to make the lists. This requires that a lot of planning has to be done on the user interface of the database system. Errors and outdated data can be reduced by better maintenance. However, the maintenance can be difficult when the signal data is scattered in several different lists. A centralized database where all signal data would be stored could make the maintenance work much easier. Another benefit of moving from spreadsheet-based lists to databases is that the newest data is always displayed for all the users of the database system. With spreadsheet-lists, one has to look for the latest version of the file.

With better maintenance and easier signal list configuration, the chances are that the signal lists provided to suppliers will be of better quality. This can help to reduce delivery times and workload, so there are possible business benefits in moving to a database system.

Signal naming and coding are some things that have been discussed a lot in the interviews of this thesis. However, this is more related to metadata that will be saved in the database. Since power plant sizes have grown, it is getting increasingly important to use more standardized signal coding and the future database should support this. In this thesis, the coding itself will not be handled in depth.

5.2 Overall picture of signal data

In the future signal database, an essential requirement is that all data should be available from the same user interface. This does not mean that the data has to physically be stored in the same location but it has to be stored in a database format and connected to a user interface where all data would be available for signal list configuration. To get a picture of all signal data available today, the different signal lists were listed with the help of Power Plants' R&D department. The signal lists were divided into three groups, standard lists which rarely change, scope related lists which depend on the project scope and project specific lists which include all signal data needed for the project. The scope related lists are mostly standard signal lists and they are used if the parts which use the signals are included in the scope. The project specific lists are what the project engineers work on in projects.

On the following pages the different signal lists which are currently available are listed. Some of the lists are marked with (SP), this shows the lists that are maintained and provided by Wärtsilä Ship Power. The content of these lists is mostly engine related signals. In these listings the MoPo-list appears twice. In the standard part the non-configured master MoPo-list with all signals is shown. The project specific MoPo-list in the project specific group, is the MoPo-list which is configured according to a particular project.

Standard lists:

- MoPo-list
- Sidex-list (engine signals) (SP)
- AVR Communication list
- Engine safety configuration list (SP)
- DC system communication list
- Flow meter communication list
- PLC to WOIS communication list
- Network monitoring communication list (under development)
- Protection relay and power monitoring unit communication list
- Frequency converter communication list

The standard lists are signal lists that normally do not change between projects. These systems that are represented by the standard lists are almost always included in power plant projects. As an example, the WOIS is almost always included and there is a standard set of signals that always go to the WOIS. In the standard lists, the engine related lists are maintained by Wärtsilä Ship Power so the maintenance of those is not included in the Power Plants' signal database.

Scope related lists:

- Heat recovery communication list
- Water treatment communication list
- Steam turbine communication list

- Fire detection communication list
- Separator communication list
- Emission treatment and monitoring signal list

The scope related lists represent systems that are sometimes in the scope and sometimes left out. For example if the plant is built as a CHP plant, heat recovery and steam turbine systems are included in the scope. This however is not concerning all plants so these systems and their signals are treated as scope related.

Project specific lists:

- Grid control communication list
- Customer connection communication list
- MoPo-list (project configured)

The project specific signal lists change in basically every project. So does the scope input from the PLM tool. In addition to these, there is a scope list with all the scope items included in the power plant project. This scope is used to filter out the project-specific signals. The generation of the communication lists is not included in this thesis but that is work that needs to be done for every project.

5.3 Generation of a project specific MoPo-list

Because the configuration of the project scope is done already in the sales tool, it is unnecessary work to do the same configuration again when generating the MoPo-list. As done in the ongoing PDM project at Wärtsilä, the sales tool data can be used to map the project scope of signals. A good solution would probably be to use a PLM system, which is currently under development, for importing the scope to the signal database system. When the parts and units that are included in the project are known, the signals connected to those parts can be listed. This however needs unique coding for the parts.

The sales tool already has this coding but from the current MoPo-list it is missing. If the coding could be realized, it could be used to compare the master MoPo-list with the generated sales scope to get a project specific signal list. One requirement for this is also that the parts and units have information of which signals belong to them. The PLM system would be an easy place to generate numerical coding for the items in the sales scope. This coding can then be matched with the signal database part coding to get a project specific list of signals.

In practice, this can be done by using XML-files. As shown in Figure 5 (page 19), the XML generated by the sales tool already contains a lot of information. The important part here is the “lineitemid” which is the unique identifier of the part or unit. Later in the signal database development it can be considered if some other parts of the metadata could be used to have some attributes ready-filled in the generated MoPo-list. This is something that was discussed in interviews with PEs as well.

Sometimes the project scope is not completely Wärtsilä’s responsibility and some other party does a part of the project. This results in that the sales scope does not include all the parts or units for the project. However, the signal data needs to be known for the complete power plant so in these part-scope cases the PEs have to add the signals manually. Because of this, it has to be possible to make a manual configuration to create the signal list. This configuration can be similar to the current MoPo-list configuration where engine type and auxiliary systems are selected.

5.4 UML models and functional requirements

To gather functional requirements for the first phases of design, use cases, activity diagrams and sequence diagrams of the main functions are drawn. As pointed out in the theoretical part of this thesis, this helps to see the main functions and to get an overview of the information system requirements.

5.4.1 Use cases

As the main roles for the signal list usage, Administrator (R&D), Project Engineer and Supplier have been identified. The R&D department takes care of the maintenance of the signal data so they have the administrator role. As the administrator of the engine signals, it has to be considered if it should just be Ship Power and the Power Plants signal database would fetch the data from their database. The PEs are doing the configuration so they work with creating the project specific signals. This includes for example the need of changing attributes and adding/removing signals in project-specific lists. The supplier does not make any changes to the list but is still an important stakeholder so that is why the supplier is included in the use cases as well. One thing to consider is if the supplier should have access to see a project specific signal list by using the software and user interface to Wärtsilä's signal database. Information security can be one concern in this case but the positive side would be that the suppliers can always see the latest version. If only a spreadsheet-file is sent to the supplier, some of today's problems might remain if newer versions of the spreadsheet-files are created.

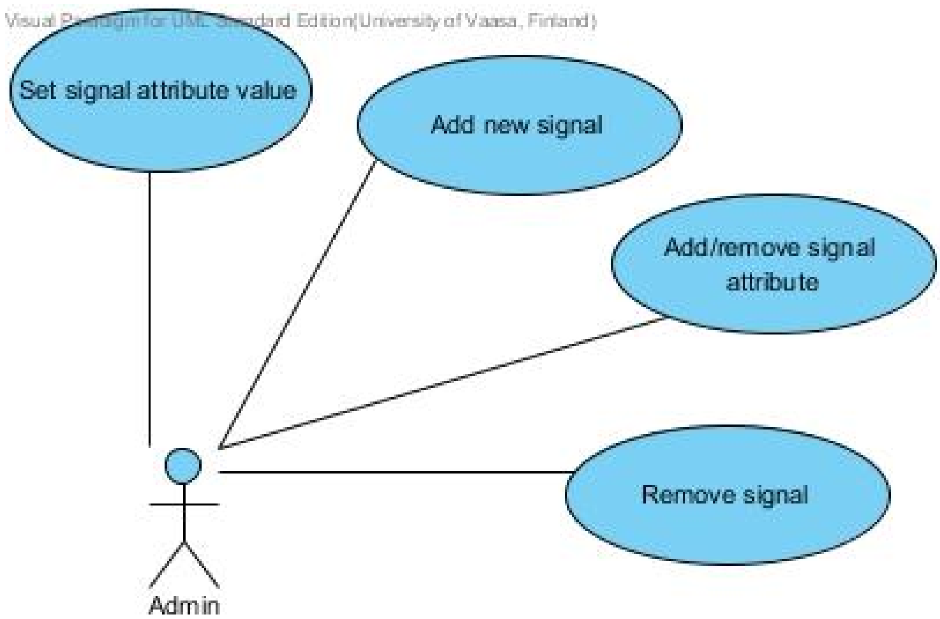

Figure 9. Admin use case.

The work of the administrator is to maintain the signal data and make sure it is up to date. This is described in Figure 9. This includes at least updating signal attributes, both changing values and adding/removing attributes. It would be good for the new database system to support adding and removing of attribute fields in case some new attributes are needed. In addition, the administrator has to be able to remove and add signals. This all would concern the master signal database.

To be able to generate a signal list for a specific project, the parts and units have to know what signals belong to them. This requires that in the database, when some signal is added or removed, information has to go to the part or unit-object to which the signal is connected. Just as in the current MoPo-list, the signals need to be shown below the part object in the user interface. In a database this has to be realized by using dependency relations. The following figure shows an example of this.

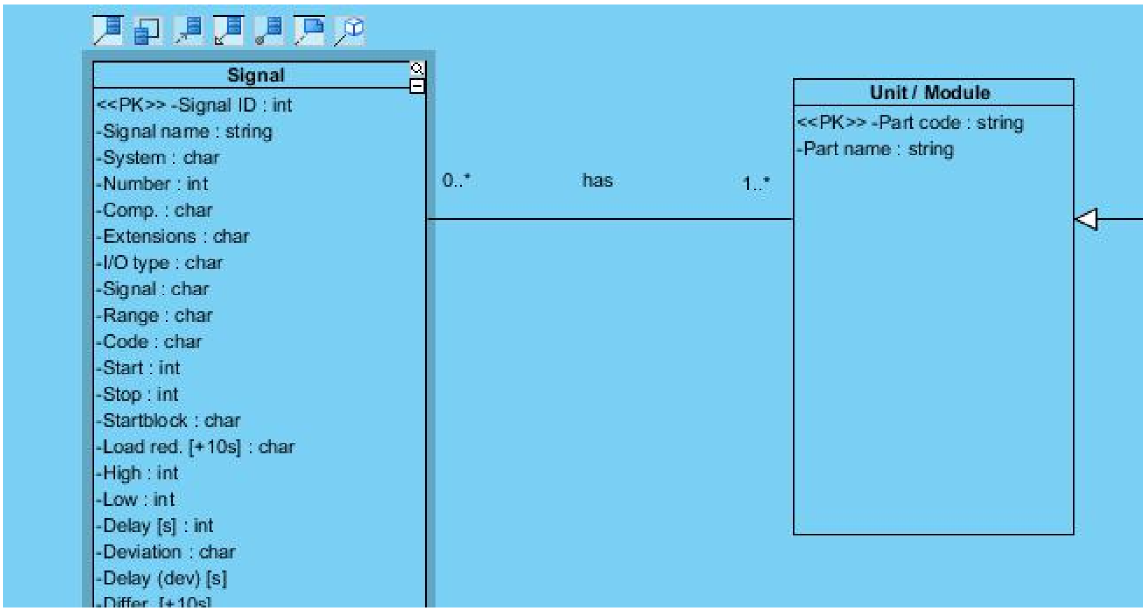

Figure 10. Associative relationship between units and signals.

The relation between the signal and the unit or module it belongs to is shown in the class diagram of Figure 10. It is a simple *has* relation where the possible amount of objects is marked in each end. A unit does not necessarily have signals but the signals always have to have something they are connected to. The same signal type can be used for different kinds of units which is why one signal can have many units associated to it. The attributes in Figure 10 are the basic attributes from the spreadsheet-version of the MoPo-list.

The project engineer is another key user of the signal database solution. Because it is often not possible to generate a final signal list from the sales scope, the PE needs to be able to add and remove signals depending on project requirements. The signal attribute values usually have to be filled in by the PE so this has to be enabled as well. The PE use case is shown in the following figure.

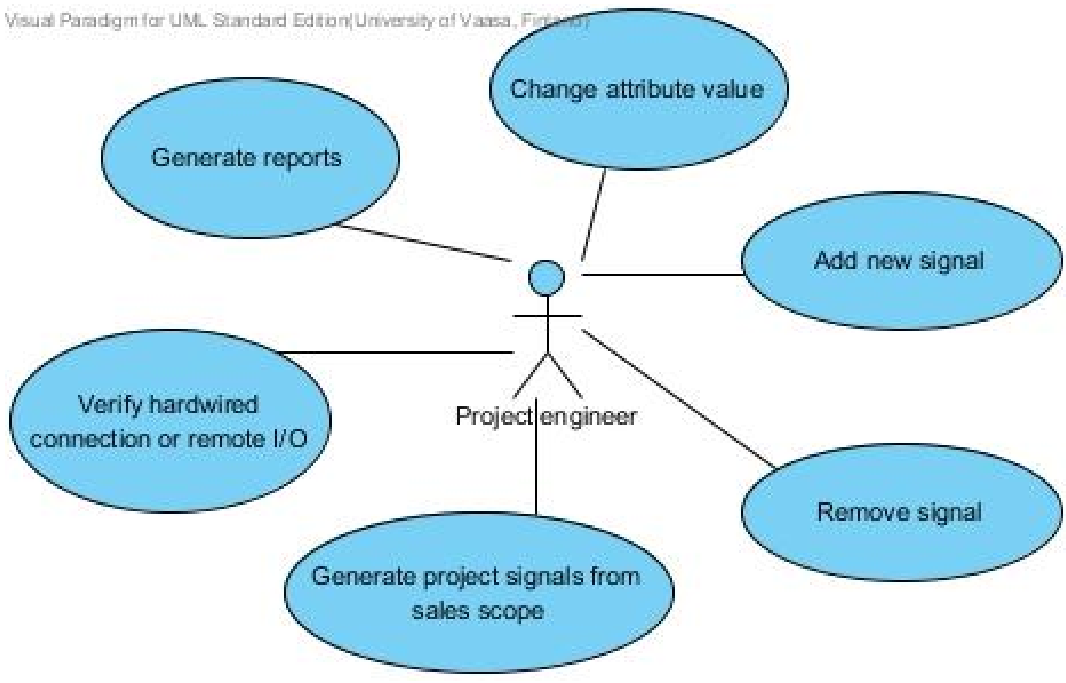

Figure 11. Project engineer use case.

Reporting is another important part of creating signal lists. It should be possible to create different reports in different file formats. Spreadsheet-reports were mentioned in some interviews. This can be one of the most easily used file formats by suppliers, even though there is no traceability and change updates in spreadsheet-files. Suppliers have also mentioned that XML could be one possible format in the future. The option of suppliers having access to some web-based user interface of Wärtsilä’s signal list database solution can also be considered. In terms of PLM, this could improve traceability and the changes throughout the project could be recorded.

5.4.2 Activity diagram

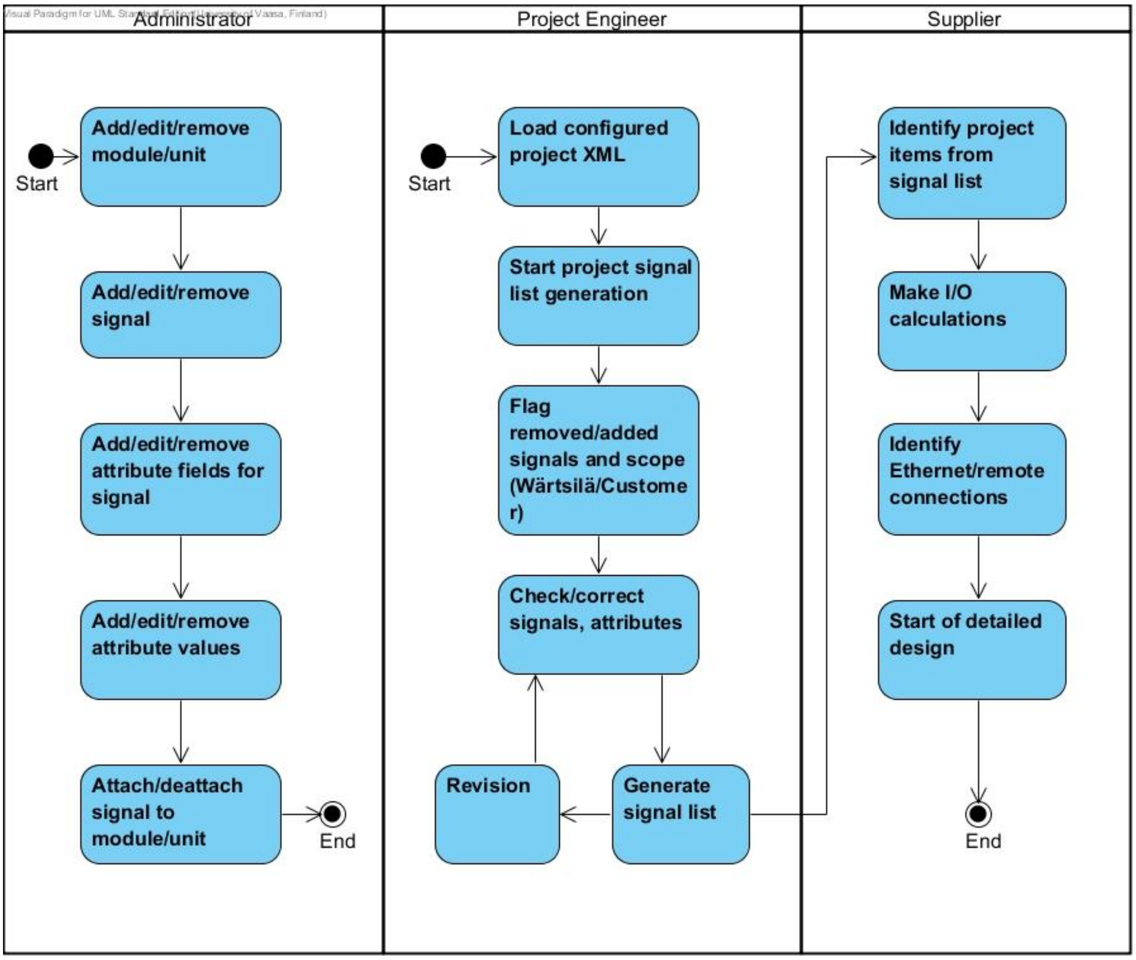

Figure 12. Activity diagram of the signal database usage.

The use cases described the tasks of single stakeholders. The activity diagram connects these use cases together and shows what different stakeholders do and in what order. The activities have a beginning and an end. In between, a process is described. A general process of maintenance, signal list creation and usage of suppliers, is described here. Figure 12 shows an activity diagram where the main stakeholders are presented. The *Administrator* tasks are a bit separated from the actual project work. The maintenance work is mainly to keep the master list updated. This can include adding or removing signals, updating attribute data and adding or removing attributes.

The tasks of PEs and suppliers are more connected together. This is shown in the activity diagram as well. When the PE has created the signal list, the list is sent to the supplier who then starts the control system design process. The example of Figure 12 shows how the project specific signal list can be generated with the help of an XML file from a project configuration tool. This does not necessarily need to be the sales configurator, other options are discussed later in this thesis. This removes the need of the PE to make the configuration again and could reduce work times. However, checking of the generated signal list is still required and for example signal ranges or attributes might not be possible to get from the sales tool, at least in some cases.

The user interface of the signal database should include the possibility to flag added or removed signals. This should at least be possible to do manually by some easy setting. Another option could be that the added or removed signals are marked automatically. Finally, the PE generates the actual list which is sent to the supplier. The data format of the list can be spreadsheet, XML or some other form that suppliers can use. As pointed out in the interviews, spreadsheets are at least still today, the most used format. The revision activity is included to highlight that there are normally many revisions of the signal list due to changing requirements during the project.

The supplier activities are also shown in the activity diagram even though the supplier is not directly interacting with the signal database. It is still good to get a picture of the supplier design process when using the signal list. Based on the supplier interviews, the most important steps in the beginning are to identify the main units included and to get the I/O calculations. The suppliers need this information to start the design of control cabinets and control software. To make the control system delivery faster, it could be considered if the I/O calculation and list of main units in the project could be sent to suppliers before all signal attribute values are finalized. This would require the possibility to generate reports with only the main units and the number of I/O signals included.

5.4.3 Sequence diagram

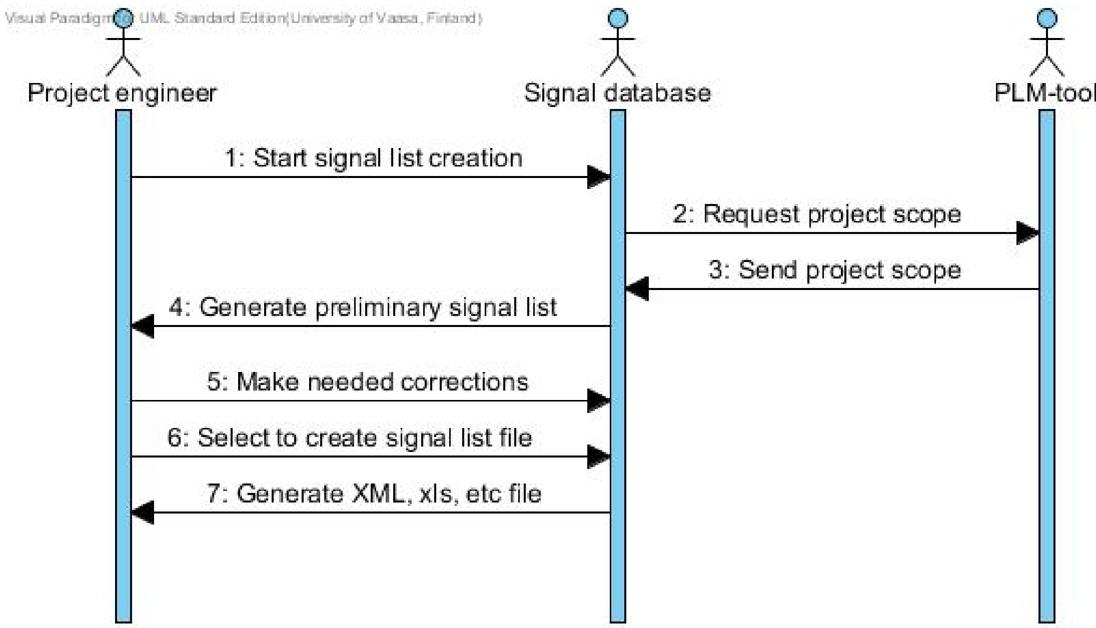

Figure 13. Sequence diagram of the signal list creation.

The sequence diagram in Figure 13 shows an example of how the information could flow between the different parties. In this case, the PE starts the signal list creation process by selecting the project for which the list is made. One solution could be that the signal database is connected to a PLM tool where the project scope information is managed. Here the software used for PLM purposes is simply referred to as the PLM tool. The database system could request the scope from the PLM tool for generation of project specific signals. After getting the scope, the project items with their signals are picked up from the master signal database by comparison with the scope. For this, a common identifier is needed both in the PLM tool data and in the master signal database.

After the comparison, the PEs will get a preliminary signal list which is based on the scope drawn from the PLM tool. However, like mentioned in the interviews, changes often have to be made so the PEs might still have to configure the preliminary list. After the final configuration is done, the PE can select to generate a file of the signal list. This file can be for example a spreadsheet or an XML file which will be sent to the control system supplier. The signal database system gives this file as an output in the final stage of the sequence diagram.

5.5 Conclusion of requirements

From the UML diagrams some requirements can already be understood. Requirements have also been gathered from interviews and current MoPo-list data. In this section a summary of gathered requirements will be presented.

5.5.1 Database content

The content of the signal database system needs to be all the signals. Not only the MoPo-list but also the other signal lists mentioned in section 5.2. This can be done either by saving all data in the same database or by connections to other databases if such already exists. This is the case regarding the engine signals for which another database is under development. However, most of the lists are some kinds of standard spreadsheet-lists so the easiest way would be to save them in one database.

The unit and module objects need to be saved in the database and they need to know what signals should be connected to them. So when a unit gets picked in the project scope comparison, the signals it has, have to get picked up as well. The interviews have brought up some suggestions about new attributes like the log-deadband for example. These were listed in the requirement conclusions of the interviews.

5.5.2 Signal list generation and file output

Because the project scope is already configured when the signal specification starts, it is unnecessary to do the configuration again. At the moment, the PLM tool where the Power Plants product portfolio is managed seems to be the best alternative to give the scope used for comparison with the master signal database data. The signal database needs to be able to receive an input file with item IDs, which can be used to pick up the corresponding items and their signals in the signal database. An option where the user could select to pull the scope from the PLM tool is something that could be considered. Otherwise the other option is to generate a report of the scope in the PLM tool and load it into the database system. It would probably be best to consider these both options already in an early stage of the signal database system development.

The types of the lists which are the output of the signal database system can be different depending on how they will be used. One example of this is the WOIS which requires an own type of data input. This input type is not defined here but for signal database output file design this needs to be taken into account. The possibility of suppliers to use XML data in the future also requires a capability of the signal database system to generate reports in XML. In normal spreadsheet-format, the layout of the spreadsheet-list is good to keep similar to the current MoPo-list. This makes it easier for users to adapt.

Logging the changes made to the generated project specific lists is important. When the list changes throughout the project, it should be possible to see what changes have been made compared to the originally generated list. This is important for tracking changes and possible errors in the list generation or project scope. For the future, it can also help the control system suppliers to get more correct signal lists right from the beginning. Today, the final list is the version according to which the control system is produced.

This is an important document for saving the status of the list at that lifecycle point. However, sometimes there are changes even at the plant site. An interesting idea is the possibility to store changes in the signal database system even in the commissioning stage at the power plant site. This would require that the project specific signal lists are saved and maintained in the signal database system.

5.5.3 Access rights

There are two main roles which will use the signal database, administrator and project engineer. The administrator role needs all access rights to modify, add and remove data. For the PE, the master data modification should not be possible. The best is to hide the administration settings for normal project engineering work. The PEs need to be able to select project scopes, perform the scope specific signal list generation and to modify the project specific lists.

For lifecycle management, additional roles might be required. For example the power plant site engineer might need to be able to correct the final changes at site to the project signal list. In addition, some other people might need access to check changes in sales scope etc.

5.5.4 User interface and configuration software

The database needs to have some kind of software running on top of it. A requirement picked from the interviews is easy usability, meaning the same kind of structure as in the MoPo-list with units having their signals listed under them. In the interviews there have been discussions about a web-browser based user interface. This interface would include options to start a new signal list project, load or pull scope information, a configuration view where units with their signals would be grouped and the possibility to select the report type and generate a report for the control system supplier. A web-based user interface would not require any software installations either and it would in that way be easier to maintain.

5.5.5 Signal list database

During the development phase of the database system, the need of a separate database for storing the created signal lists came up. This is needed especially in revision control. New revisions of the signal lists have to be possible to save. When the PE gets the first version done, the list is normally sent directly to the control system supplier. However, from the interviews it has been noticed that there are often changes to signals or even units/modules. This results in newer versions of the signal list which are revisions from the first one. For traceability reasons the older revisions need to be saved, this enables people to see what has changed during the project.

It is good to notice that the signal list will not only exist during the power plant project, but it will also remain as a document to describe the As-built situation. When saving the revisions in the project specific signal list database, the project ID needs to be connected to the list as some kind of metadata, which would enable to search for project specific lists after the plant is built and the project is closed.

5.6 Possible solutions

This section describes a possible overall solution which could be recommended based on the results of the interviews and the literature review. The first thing to consider is the actual database itself. Things like who will develop it and who will maintain it have to be decided. There are already many lists maintained by the R&D department at Power Plants. Because of this, it would be best to continue with the maintenance at the R&D department.

5.6.1 Development responsibility

The resources for developing the database itself within Wärtsilä might be limited, so for an advanced solution with good possibilities for eventual future development and system connections, it could be a good idea to outsource the development part. The database content itself is not so complicated, at least not the MoPo-list signals which have been the main focus of this thesis. The other signal lists might bring more requirements which have not yet been considered.

5.6.2 Database model

There are basically two object classes in the database, signals and units/modules. The units and modules know which signals belong to them. When the unit/module gets selected in the project scope comparison, the signals belonging to that unit/module will also be selected. In the maintenance view of the database system, the relations can be used to show all related signals when a unit/module is selected. Even though several units or modules might have the same signal, it would be better to have own signals for each of them. This is because of possible changes in signal information that would be needed in one module. It should not be allowed that changing values for one signal would change signal information in all units or modules which use the same signal type. If the user wants this, it could be done with a separate function. A possible solution for the same kinds of signal types could be sub-classes of signals. For example analogue and digital signals could have their own sub-classes which are based on the general signal class. The sub-classes can help in the maintenance work because it would be easier to manage certain signal types.

5.6.3 Project specific signal list generation

The project specific signal list could be generated either automatically by clicking a button, or then the user would have to load a file into the signal list configuration system.

The project scope will most likely be taken from an existing PLM tool which can generate different output files, for example spreadsheet-files. It would probably be possible to pull a project scope from the PLM tool when knowing the project ID. However, this pulling function can require a lot of work before the connection is set up. This is why it should also be possible to load a scope file in manually, for example in spreadsheet or XML format.

To get the project specific signals, both the units/modules in the signal database and in the project scope from the PLM tool need to have a comparable ID. In the PLM tool it is easy to add an item detail which contains the ID for this purpose. The ID can just be a running number. After this, the units and modules in the signal database have to be mapped to this ID. When the mapping is done, it is easy to make a comparison between the project scope and signal database to pick up the units and modules included in the project. When the units and modules get picked, the signals which have a relation to them also get picked. These signals can be graphically shown under the unit/module.

5.6.4 Updates to project or signal data

During the project, there are often changes to the project scope because of new requirements. Changes because of updates and maintenance can also happen in the signal setup of the units and modules in the signal database. Especially notifications of scope updates are important for the project engineers, otherwise the signal list might not match the project scope and all the parts and modules included. However, it would also be good to have notifications of changes in the master signal database. For usability, a good solution would be a button in the user interface which would check if there have been any changes to the project scope or signal database after the initial signal list has been generated. This would require a connection between the PLM system and the signal configuration system. If this connection does not exist, the best solution would probably be to inform project engineers about scope changes so they can load a new scope file into the signal configuration system and make a new comparison. However, this would not create a new list, it would only highlight the scope changes in the signal list under work. The data in the signal database would then be easier to check. In this case the “Check for updates”-button would be the best solution.

5.6.5 User interface

The user interface solution would be based a lot on the requirements in section 5.5. The signals can be grouped under the units or modules. In the current MoPo-list, the units and modules are also grouped into systems, like fuel system for example. To improve the usability from the current MoPo-list, these groups and units/modules should be possible to expand or hide. So when the “Fuel System” is clicked, all the units/modules belonging to the fuel system would expand below. The same would apply when clicking a unit or module, then the signals would expand under them.

Some functionalities that should be included in the user interface are:

- Loading of project scope file.
- Making the comparison between the scope file and the master signal database by clicking “Generate project specific signal list”.
- As an alternative to scope loading, a pull function that would automatically get the scope from the PLM system.
- Possibility to change the attribute values of the signals.
- Possibility to remove or add a signal.
- Flagging of added or removed signals.
- Marking of optional signals.
- Possibility to save different revisions of the signal.
- Option to release signal list to IDM (Internal Document Management).

In addition to a functional grouping, a grouping by area would be useful, this can help PEs to see where different signals should be connected to avoid too much cabling.

More things to include in the user interface can be found in Table 1 (page 44). There, some signal attributes and other details of signals that were suggested by PEs, R&D people and suppliers, are shown.

6 RESULTS

This chapter will overview the main results and analyzes the work done in requirements gathering and system modelling. As in the system development lifecycle presented by Tegarden et al. (2013: 9), this thesis includes the planning and analysis phases. In chapter 4 the requirements were gathered from different stakeholders. In chapter 5 the requirements were analyzed and a functional description of the system was made by using UML models.

In the interviews with R&D, project management and suppliers, it was clearly shown that there is a business need for a more advanced and modern way to manage the power plant signal configuration. The data maintenance is difficult, connection to other information systems is poor and the lifecycle management of the signal configuration could be improved. As the interviews showed, this can show as improvements in quality and delivery times. These are two of the three main operational performance measures, the third one being cost. From the stakeholder interviews it can be said that the business justifications, which Tegarden et al. (2013: 4) and Sommerville et al. (1997: 63−111) see as basic parts of the initial planning of the system development, do exist. In the requirements gathering part, many reasons of why a better signal configuration tool is needed, came up.

The analysis part of this work collected the requirements gathered from the interviews and presented a functional description of the power plant signal configuration process. A high-level system concept of the planned (To-be) system was drawn to see connections to other information systems used by Wärtsilä. These connections are essential to make a functioning signal configuration tool which can re-use data already available in other information systems. This analysis part presented the *Analysis* phase of the system development lifecycle by Tegarden et al. (2013: 9), the *Software Requirements* and *Analysis* parts of the second and third step in the waterfall model shown by Guntamukkala et al. (2006) and the first three steps in the RUP system development model presented by Sommerville (2011: 50−52).

Table 2. List of requirements with requirement type included.

Role	Requirement	Type
R&D/ Suppliers	International coding standard	Data requirement
R&D/ Suppliers	Standardized signal naming	Data requirement
R&D	Alarm priority groups	Data requirement
R&D	Integration between engine signals and power plant signals	Functional requirement
R&D	An attribute fir the *high high* -alarm	Data requirement
R&D	I/O calculation to be more clear (not only check if *Connect to PLC* is filled)	Reporting requirement
R&D	*Trend* possible to fill only for analogue signals	Data requirement
R&D	Possibility to get an *As-built* signal list	Reporting requirement
Project engineers / Suppliers	Unique signal coding	Data requirement
Project engineers	Newest MoPo-list version easily available	Functional requirement
Project engineers	Spreadsheet-file as output type	Reporting requirement
Project engineers	Possibility to make corrections/changes in the GUI	Functional requirement
Project engineers	Optional signal available and marked	Data requirement
Project engineers	Possibility to configure the complete the MoPo-list manually	Functional requirement
Suppliers	Unique tag codes	Data requirement
Suppliers	Attribute for log-deadband	Data requirement
Suppliers	PLC-signal blocks included in the MoPo-list	Data requirement
Suppliers	Both short and long signal description text	Data requirement
Suppliers	Unique signal naming	Data requirement
Suppliers	Marking of customer / Wärtsilä scope	Data requirement
Suppliers	“x” values need to be filled by PEs	Data requirement
Suppliers	Clearer attribute descriptions	Data requirement
Suppliers	Signal list to match with the project flow chart	Data requirement
Suppliers	An attribute for connection to the plant external communications	Data requirement
Suppliers	A *General fault* attribute	Data requirement

From UML, the use-case, sequence and activity diagrams were used to model the signal configuration process. This helps to see what tasks are included in the configuration work. The signal database system needs to have these functionalities included. From the interviews and requirements gathering it can be concluded that an object-oriented model based on the OOSAD methodology shown by Tegarden et al. (2013: 28−29) is well suitable for the signal database. Both the signal and the modules/units which they belong to, can be modelled as objects with certain attributes. The modules/units will have a relation to their signals so they know what signals are connected to them.

The requirements gathered from the interviews were already listed and grouped by stakeholders. There are also other ways how the requirements can be grouped. Chemuturi (2013: 5) writes that one way is to divide them according to functionality, product considerations or source considerations. Conger (2012: 47) groups them by data, reporting and security. In Table 1 (page 44) the requirements were grouped by source consideration as Chemuturi (2013: 5) presented. This source consideration grouping is aligned with the activity diagram of Figure 12 (page 53). It can clearly be seen what requirements there are in the different roles of the activity diagram. The use-cases are also grouped according to stakeholders.

In Table 2 the requirement types presented by Conger (2012: 47) have been added. These are however not sufficient for this type of system development. To get a requirement type for all requirements, a *functional* requirement type has been added. This is not directly related to data content or reporting, that is why there is a need for this type. The *functional* requirement type describes functionalities that the system needs to be able to perform to do the required tasks.

As it can be seen from Table 2, there are no access or security requirements listed. This is something that was not discussed much in the interviews. The reason to this is partly because the usage of the signal database will concentrate inside Wärtsilä and there are mainly two roles: data maintenance and project work. More precisely, one role is database administration and the other role is project engineering. These are already described in sub-section 5.5.3. Besides from the access rights, the overall security of the database must be considered. For example if it should be located on an internal server or if some cloud service can be used. Cloud services might be faster to deploy and possibly cost saving, however it needs to be discussed how sensitive the data in the signal database is.

UML models of the functional descriptions were drawn in this thesis. UML supports the OOSAD development model presented by Tegarden et al. (2013: 28−29) and it describes the main functionalities. UML could also be used for more detailed system modelling but for now, the model of Figure 8 (page 43) is enough to describe the connections. For the requirements part, the UML models used in this work, are enough. Based on the requirements a detailed system model can be developed.

One thing that has been brought up in discussions of the database development is how the master database of signal data is synchronized with the product portfolio. This is something that is still missing from Figure 8 (page 43) and it needs to be discussed further. If there is no linkage, the result is that the product portfolio is maintained in two different places. One solution is to import the product data to the signal database and after this import add the signal information. This could keep the signal database synchronized with the product portfolio. Another challenge is to integrate all lists presented in section 5.2. The problem is that the signal attributes might vary from the MoPo-list. A solution that is used in modern data management system for this problem is user-defined attributes. This basically creates object types for objects from different lists. These objects have their own pre-defined attributes that should be filled. This system modelling is based on the object-oriented thinking which is common in many modern IT systems. However, it requires much more from the signal database system compared to standard data attributes. But if one wants to get all the benefits of the new database, all signal data should be included.

The purpose of this thesis was to find the main requirements and to develop solution concepts for a signal database. Through interviews, the main requirements have been gathered from all main stakeholders. During the requirements gathering process, many new aspects were found. For example all the other signal lists in addition to the MoPo-list. The connection to PLM is also something that got clearer during the writing of this thesis. A suggested data model for the signal and module/unit objects was made to give a picture on how the database can be modelled. The solution concept part was limited to a high level picture which still has some open questions, for example the synchronization with the product portfolio. The requirements on the other hand, were focused a lot on usability and to make the project and maintenance work easier. Since the maintenance and data quality were some of the main reasons to plan this new database, the requirements should give an important input to the development work.

The scope of the database content has extended during this development work. More different list and integration possibilities have been found. The signal database that could bring full benefits is now larger than what was originally thought. This is good to know when considering investment costs and feasibility. Some of the wishes from PEs and other stakeholders might not be possible to fulfil but most of them were related to data quality issues which should be possible to solve. Some positive findings include for example an almost finished similar database on Wärtsilä Ship Power side and clear support from all stakeholders for the new database system.

The future work should start with the decision of continuing the development. Even though the scope of the database is a bit larger than planned, many clear benefits have been brought up in this requirements analysis work. From this point it would be good to have the technical database development team included to discuss how the requirements can be fulfilled in the proposed system. For this, a vendor or an internal team needs to be chosen.

Based on the findings of this work, some alternatives for the database system can be recommended. The best suitable way to realize the database could be:

1. A cloud-based platform which enables own configurations. This would be a Platform as a Service (PaaS) type of cloud server where the base software is provided.

2. An internal server where the same model of Ship Powers signal database would be used to avoid starting the work from the beginning.

Both of these should be quite cost efficient and the development times should not be too long. In PaaS cloud services there is an existing platform with base functionalities. However it is not a fully functional software for signal data management, it serves more as a base for users to build their own configurations on. This could be used to define own signal groups, part relations and signal attributes. On the other hand, the Ship Power business unit of Wärtsilä already has an almost ready database system for their signals. This could save a lot of development time if it is suitable for power plant signal configurations as well.

Considering option (1.), the possible benefits would be that the configurations of the software can be done by internal people, flexibility of software properties and less dependency of externals. Drawbacks might be the need of external IT experts for connection to other database systems, resources are needed for configuring the software and it might be difficult to find a PaaS service with all the required properties. In off-premise cloud solutions the data security can also be an issue. In option (2.) the benefits are that a lot of the work is already done and compatibility with the Ship Power signal database. Drawbacks might be compatibility with other information systems, suitability for power plants signal configuration and dependency on external database developers.

At this point it would be good to search for different available PaaS services and/or take a closer look on how the signal database of Ship Power would be suitable for power plant signals. Besides these two suggestions, there is always the possibility to find a new external database developer or to use Wärtsilä Information Management services.

In addition to the requirements of Tables 1 and 2 (pages 44 and 65), general requirements like system speed, easy usability, access policy, security policy and compatibility with other information systems need to be remembered. The access policy should not be too complicated to cause long waiting times for the right people to get access. When choosing servers or cloud providers, the speed has to be considered because people do not want to wait long times during project specific signal list generation for example. Good usability results in a more positive attitude among users and they are therefore more interested in developing and giving feedback.

7 CONCLUSIONS

The objective of this Master's thesis was to find requirements and develop a concept model for a monitoring and control signal database of Wärtsilä power plant signals. The database development work started because of maintenance challenges and a large manual workload of the current spreadsheet-based signal data. Interviews were performed with the main stakeholders of the signal lists. These are the R&D department, project management and control system suppliers. In the interviews the main problems today and the future requirements were discussed. The development process has been viewed from information system engineering, requirements management and electrical engineering points of views. In addition, supplier information exchange has been considered. All these views can give a good overall picture of how to continue the development.

Based on the results of the interviews and findings from literature, the main requirements were listed and classified. A functional description of the proposed signal database was developed with the help of UML models. Finally, a high-level concept model of the proposed system set-up was created. In the work process, parts of different software development models, including the waterfall model and the OOSAD model, were used as a base for the requirements management and analysis parts in the development process.

The requirements study showed that the scope of the signal database is larger than it was first thought and revealed more signal lists that should be included. The connections to other information systems are clearer and a high-level description is made with the help of UML. The interviews show the need of a new signal database and based on the requirements, solutions like PaaS services or usage of similar internal database systems have been presented to be a platform for the planned signal database.

REFERENCES

Brace, William & Vincent Cheutet (2012). A Framework to Support Requirements Analysis in Engineering Design. *Journal of Engineering Design* 23:12, 873−901.

Chemuturi, Murali (2013). *Requirement Engineering and Management for Software Development Projects*. New York: Springer. 266 p. ISBN: 978-1-4614-5376-5.

Conger, Steve (2012). *Hands-on Database: An Introduction to Database Design and Development.* Upper Saddle River, New Jersey: Pearson. 195 p. ISBN: 978-0-13-610827-6.

Guntamukkala, Vamsidhar, H. Joseph Wen & J. Michael Tarn (2006). An Empirical Study of Selecting Software Development Life Cycle Models. *Human Systems Management* 26 (2006), 265−278.

Hadaya, Pierre & Philippe Marchildon (2012). Understanding Product Lifecycle Management and Supporting Systems. *Industrial Management & Data Systems* 112:4, 559−583.

IBM Technical Library (2014). *UML basics: An introduction to the Unified Modeling Language.* Accessed 20.1.2014. Available from the World Wide Web: URL:< http://www.ibm.com/developerworks/rational/library/769.html>

Lindsley, David (2000). *Power-Plant Control and Instrumentation – The Control of Boilers and HRSG Systems.* London, UK: The Institution of Electrical Engineers. 222 p. ISBN: 0-85296-765-9.

MoPo-list and Process Drawing Balancing. 2014. Training session notes. Vaasa, Finland. 7.3.2014. Wärtsilä internal training session.

Naiburg, Eric J. & Robert A. Maksimchuk (2001). *UML for Database Design*. Upper

Saddle River NJ: Addison-Wesley. 300 p. ISBN: 0-201-72163-5.

Power Plants Solutions (2013). Marketing Material. Accessed 23.9.2013. Available from the World Wide Web: <URL: http://wartsila.fi/file/Wartsila/en/1278518335887a1267106724867-Power-Plants-Solutions-2013---3rd-Edition.pdf>.

Ritchie, Colin (2008). *Database Principles and Design*. 3rd ed. London: Cengage Learning. 388 p. ISBN: 978-1-84480-540-2.

Saaksvuori, Antti & Anselmi Immonen (2002). *Product Lifecycle Management.* Heidelberg, Germany: Springer. ISBN: 978-3-540-78173-8.

Sales Tool Introduction (2013). Internal training material. Accessed 3.10.2013. Unpublished. Wärtsilä Internal Network.

Schuh, Günther, Henrique Rozenfeld, Dirk Assmus & Eduardo Zancul (2008). Process Oriented Framework to Support PLM Implementation. *Computers in Industry* 59 (2008), 210−218.

Senior Chief Project Engineer, Electrical, Wärtsilä Power Plants & Project Engineer, Electrical, Wärtsilä Power Plants. 2013. Interview. Vaasa, Finland. 26.11.2013.

Senior Project Engineer, Electrical, Wärtsilä Power Plants. 2013. Interview. Vaasa, Finland. 10.9.2013.

Signal Database Meeting, Wärtsilä Ship Power & Power Plants. 2013. Meeting notes. 30.9.2013. Vaasa, Finland.

Sommerville, Ian & Pete Sawyer (1997). *Requirements Engineering: A Good Practice Guide.* West Sussex, England: Wiley. 391 p. ISBN: 0-471-97444-7.

Sommerville, Ian (2011). *Software Engineering.* 9th ed. United States of America: Pearson. 773 p. ISBN: 978-0-13- 705346-9.

Supplier 1. 2013. Interview. Vaasa, Finland. 26.11.2013.

Supplier 2. 2014. Interview. Vaasa, Finland. 21.1.2014.

Tegarden, David, Alan Dennis & Barbara Haley Wixom (2013). *System Analysis and Design with UML.* Singapore: Wiley. ISBN: 978-1-118-09236-1.

Wärtsilä Elwis (Electrical Wisdom) (2013). Accessed 24.9.2013. Unpublished. Wärtsilä Internal Network.

Wärtsilä Power Plants Electrical & Automation-team Meeting. 2014. Meeting notes. 23.1.2014. Vaasa, Finland.