CODEN:LUTEDX/(TEIE-5479)/1-54(2022)

Industrial Electrical Engineering and Automation

A General SCADA Application for Water and Wastewater Treatment

Markus Kvist
Johan Vikstrand

Division of Industrial Electrical Engineering and Automation
Faculty of Engineering, Lund University

A GENERAL SCADA APPLICATION FOR WATER AND WASTEWATER TREATMENT

LUNDS UNIVERSITET
Lunds Tekniska Högskola

Division of Industrial Electrical Engineering and Automation, IEA

Master Thesis, EIEM01

Master Student
Markus Kvist
Markus.love.kvist@gmail.com

Master Student
Johan Vikstrand
Johan.wikstrand@telia.com

Company Supervisor
Mattias Nilsson
Mattias.nilsson@kentima.se

Academic Supervisor
Gunnar Lindstedt
Gunnar.lindstedt@iea.lth.se

Examiner
Ulf Jeppsson
Ulf.jeppsson@iea.lth.se

Abstract

The average Swede uses 140 liters of water everyday. This combined with the usage in industry means that Swedish municipal water treatment plants produce approximately 900 billion liters of clean and safe drinking water every year. This does not include the fact that all water flowing to a treatment plant needs to be treated before being sent back out into the environment. These facts hint at what a huge scale these processes operate at. Furthermore this needs to happen every hour of every day, 365 days a year. So how can these processes be monitored to ensure that everything works as intended and any problems are found and resolved as quickly as possible? With the help of a Supervisory Control And Data Acquisition (SCADA). In simple terms a SCADA system is a way to supervise and manage a process.

Kentima is a Swedish company located in Staffanstorp, Skåne, Sweden. They specialise in developing and selling products for the automation and security sector. One of the software sold by Kentima is WideQuick. WideQuick is a software suite which enables the creation of Human Machine Interfaces (HMI) and SCADA solutions. It is on the behalf of Kentima that this project was started as they want to expand their services with a specialised SCADA for the water and wastewater industry.

The scope of this thesis is to lay the foundation for a general SCADA solution that can quickly and easily be adapted to function with any water and wastewater facility. The requirements for the functionality of this were generated through an academic research survey. This was complemented with interviews from customers active in the field of water and wastewater with varying responsibilities, such as system integrators. The combination of the latest academic research with the knowledge from people in the industry was thought to give the best possible application. This is because it would give a good balance between the best practices from the academic world along with the practices that are actually used in the industry.

As mentioned earlier this project lays the foundation of a SCADA solution. This includes a navigation bar to navigate between different processes. An object library to represent different objects in the process, as well as object control, analysis and overview. The thesis will explain how this was achieved and the reasoning behind the choices made. After the application was build it was then demonstrated to both the company and the customers. This was to get their input and feedback on what looks good, what needs to changed and to inspire things to be developed in the future.

The final part of the thesis primarily discusses the feedback from the customers. It also lays out suggestions on things to further develop in the future to complement the project into a complete SCADA application for the water and wastewater industry.

Sammanfattning

I snitt förbrukar varje svensk 140 liter vatten om dagen. I samverkan med användningen inom industrin innebär att kommunala vattenverk producerar 900 miljarder liter rent och säkert dricksvatten varje år, detta inkluderar inte det faktum att allt använt vatten måste behandlas innan det kan skickas tillbaka ut i naturen. Detta visar på de enorma skalor som dessa processer arbetar på. Dessutom behöver detta fungera varje timme, varje dag, 365 dagar om året. Så hur kan dessa processer övervakas för att försäkra att allting fungerar som tänkt och att alla problem kan hittas och lösas så fort som möjligt? Med hjälp av ett "Supervisory Control And Data Acquisition (SCADA) system. I enkla drag så är SCADA ett sätta att övervaka och kontrollera en process.

Kentima är ett svenskt företag beläget i Staffanstorp, Skåne. De specialiserar sig på att utveckla och sälja produkter för automations- och säkerhetsbranchen. En utav mjukvarorna som säljs av Kentima är WideQuick. WideQuick är ett mjukvaruportfolio som möjliggör skapandet av Human Machine Interface (HMI) och SCADA lösningar. Det är på ett initiativ av Kentima som det här examensarbetet påbörjades då de vill utöka portfolion av applikationer med en specialiserad SCADA lösning för vatten- och avloppsindustrin.

Omfattningen av detta examensarbetet är att lägga grunden för en generell SCADA lösning som snabbt och enkelt kan anpassas för att fungera med alla vatten- och avloppsverk. Kravspecifikationen för funktionaliteten av detta skapades genom en akademisk studie av akademiska arbeten. Detta kompletterades med intervjuer av kunder aktiva inom vatten- och avloppsfältet med varierande ansvarsområde, så som systemintegratörer. Tanken bakom kombinationen av akademisk forskning tillsamans med kunskapen från folk inom fältet var att skapa den bästa möjliga applikationen. Detta eftersom det innebär en blandning av de bästa teknikerna från den akademiska världen tillsamans med de tekniker som faktiskt används inom industrin.

Som nämndes tidigare kommer detta examensarbete lägga grunden för en generell SCADA applikation. Denna inkluderar en navigationsmeny för att navigera mellan de olika processerna. Ett objektsbibliotek för att representera olika objekt i processen, samt objektskontroll, -analys och -översikt. Examensarbetet kommer förklara hur detta har åstadkomits samt förklara de beslut som togs. Efter att applikationen byggdes så demonstrerades den för både Kentima och deras kunder. Detta var för att få deras åsikter om vad som ser bra ut, vad som behöver ändras och deras behov för framtiden.

Den sista delen av rapporten diskuterar både feedback från kunder och presenterar förslag på saker att utveckla i framtiden för att komplementera projektet till en fullständig SCADA lösning för vatten- och avloppsindustrin.

Acknowledgements

We would like to thank Kentima for allowing us to take on this very exciting project. Furthermore, we would also like to thank them for supplying everything needed to complete the project, including but not limited to license keys for the various software as well as putting us in contact with customers active in the water and wastewater industry. A special thanks goes out to our company supervisor Mattias Nilsson for helping us with both the technical and design aspects of this project.

We also want to thank our academic supervisor Gunnar Lindstedt and our examiner Ulf Jeppsson at Lund University for supporting us throughout this project.

Lastly we want to dedicate this thesis to the memory of Fabian Kvist. Although no longer with us, his memory will live on forever.

Division of Labour

The code for the application was written using pair-programming i.e two programmers on one computer. One of the reasons for this was that this code lays the foundation for the program and any design decision made now will affect the application for many years to come. Therefore it was decided that writing the code together would ensure consensus concerning the application structure. The same goes for the design of the application.

The thesis report was also written together, however Johan focused more on parts concerning HMI and SCADA while Markus was responsible for the water and wastewater aspect.

Contents

Acronyms

Acronym	Full name
BMS	Building Management System
DBMS	Database Management System
GUI	Graphical User Interface
HMI	Human Machine Interface
HVAC	Heating, Ventilation and Air Conditioning
I/O	Inputs and Outputs
KPI	Key Performance Indicator
MTU	Master Terminal Unit
ODBC	Open Database Connectivity
OPC	Open Platform Communication
P&ID	Piping and Instrumentation Diagram
PLC	Programmable Logic Controller
PSIM	Physical Security Information Management
RS-xxx	Recommended Standard followed by number of standard
RTU	Remote Terminal Unit
SCADA	Supervisory Control and Data Acquisition
SQL	Structured Query Language
TCP/IP	Transmission Control Protocol/Internet Protocol
UI	User Interface
WWTP	Wastewater Treatment Plant

"A picture is worth a thousand words. An interface is worth a thousand pictures"

-Ben Shneiderman American Computer Scientist

1 Introduction

This thesis handles the development of a general SCADA application focused towards water and wastewater treatment. This chapter introduces the topic and background information related to the thesis.

1.1 Background

The primary aim of treating water and wastewater is to care for the environment around us and provide the public, industries and businesses with clean drinking water as well as water for other uses. According to the World Health Organisation (WHO), 54% of the global population have access to safely managed sanitation services and 45% of household wastewater is discarded without safe treatment[1].

Traditionally managing a water and wastewater treatment plant required large staffing, time and resources. To improve efficiency in day to day operations a Supervisory Control And Data Acquisition system, a SCADA system, can be applied to the process. A SCADA system provides the user with real-time data and analytic tools to get an overview of the process to detect leaks, overflows and chemical imbalances as well as alerting operators of these unwanted behaviors in the process. It does this by collecting process performance data such as flows, pH-levels, temperatures, oxygen levels and so on.

Kentima is a Swedish company that was created in the year 2000 in Staffanstorp. The company develops, manufactures and sells products for the automation and security sector. It has a wide range of products such as Human Machine Interface software, HMI, SCADA, Industrial Computers, Video Management Software, VMS, as well as Operator panels and -boxes.

Currently Kentima provides a general SCADA application called WideQuick Building Managment System, BMS. BMS can be used for managing building, heating, ventilation and air conditioning, also known as HVAC. The application provides system integrators with tools to quickly create an overview of a facility with an assortment of smart objects and analytical tools for its monitoring and visualization requirements. WideQuick BMS has also previously been re-purposed by organizations in the water and wastewater sector.

As a further expansion of applications provided by Kentima AB such as the WideQuick BMS, a similar application for water and wastewater treatment will be developed in this thesis. This will include with smart objects, easy to configure navigation and a simulated example plant.

1.2 Objective

The objective of this project is to develop the foundation of a competitive general SCADA application focused on functionality tailored especially for water and wastewater treatment. The application should simplify the generation of unique objects to satisfy the industry's varied plant/process implementations, shorten the time on commissioning a plant design thereby reducing time to market for the system integrator. The application should also provide the end user with useful information in an intuitive way so to simplify supervising and managing the complexity that the water and wastewater treatment processes entails.

1.3 Software

In this project a collection of different tools will be used. A short description of these are given here.

The WideQuick Suite

The WideQuick suite consists of WideQuick Designer, WideQuick Runtime, WideQuick Remote and WideQuick Web Client. WideQuick provides an embedded OPC-client and MODBUS protocols and therefore WideQuick can communicate with a large number of I/O, PLCs and other equipment.

WideQuick Designer is the developer environment where a user designs the plant or system that they want to monitor. Configuration of databases, schedules, tags to monitor and scripting can also be done with the Designer software.

WideQuick Runtime runs the application that the user would run on a HMI-panel, Linux or Windows machine as well as on a server. In other words this is the application that an operator in the plant or control room interacts with.

WideQuick Remote enables one or more users and projects to connect to a target system remotely. The software enables configuration of such systems.

WideQuick Web Client allows an operator access to the SCADA application built with WideQuick Designer through any web browser. Web Client handles updating the web GUI and how often data should be synced with the WideQuick Runtime application.

KepServerEX

KepServerEX from KepWare allows the user to connect, manage and control signals through a central user interface. The software will be used in this project to simulate OPC and MODBUS tags. Functionality provided in KepServerEX for simulating tags allows for signals to be represented either as wave functions or predefined patterns [2].

ModRSsim 2

ModRSSim 2 is a Modbus RTU and TCP client simulator. It is used for simple testing and simulation of MODBUS tags. Similar to KepServerEX this simulator also allows for signals to be user defined, however ModRSsim 2 signals are coded using the Visual Basic programming language.

MKDocs

MKDocs is a open-source static site generator geared towards building project documentation [3]. It is used to document use cases for the application. It is here that the users are able to read about objects built for the object library and scripts available in WideQuick Water and Wastewater. MKDocs was chosen for its easy to read, easy to write syntax as well its auto-generated search functions.

Visual Studio Code

Visual Studio Code is a lightweight code editor with support for JavaScript and an assortment of other languages and extensions. These extensions provide functionality that speeds up the work process of writing code. The code editor will be used for developing larger scripts in the SCADA application.

MySQL Workbench 8.0

MYSQL Workbench is a visual tool for database design that integrates administration and development of database design. For this project it is used for handling the relational databases that log performance data of a system or plant.

1.4 Limitations

As this thesis project aims to be the foundation of Kentima´s Water and Wastewater Treatment application it will contain many but not all features of a complete SCADA-solution. Therefore the focus in this project includes but is not limited to:

- An Object Library
- Navigation
- Application architecture
- Object specific popups
- An example plant with simulated tags

Outside the scope of this project are more in-depth analytical functions, such as built in report tools and messaging services. Nor is the cyber security aspect application taken into consideration since this is primarily handled by the WideQuick developers. Additionally, due to time constraints the application is only tested in WideQuick Runtime and not with WideQuick Web Client or -Remote.

2 Literature review

In this chapter a brief introduction of technologies, processes and best practices needed to conduct the project are presented.

2.1 Wastewater Treatment

The goal of a wastewater treatment plant or WWTP is to process water from different sources and remove its pollutants. This is generally done in four or five steps [4][5].

1. Pre-treatment or mechanical treatment is where solids and larger particles are removed from the water. This is typically done by filtering the water through grates of decreasing sizes which removes the larger particles. This is often followed by a sand filter, where the water is allowed to sit or flow slowly through pools which in turn causes smaller particles to settle at the bottom [5].
2. The next step is biological treatment. In simple terms the water is treated through different processes to encourage the growth of "good" bacteria. The bacteria in turn consume unwanted or harmful pollutants and break them down into harmless ones.
3. After the biological treatment comes the chemical treatment. This involves adding different chemicals to the water which in turn then cause different chemical compounds, such as phosphorus which are solvable in water to solidify and precipitate to the bottom.
4. Secondary sedimentation is the next step in the process which involves collecting all of the waste and the sludge from the treatment process and separate it from the purified water.
5. Finally the sludge can then be collected and dried through several different processes. The sludge can be used for a variety of things, such as creating bio-gases or be incinerated.

2.2 Water Treatment

There are primarily two ways to collect water to be supplied as fresh water. The first is to use surface water. The other option is to pump and collect groundwater [6]. Due to a more limited supply of groundwater it is less suitable for larger cities.

Groundwater

The quality of groundwater in Sweden is often high [6]. Since the water does not contain dirt (humus) the water is generally clear. The water does however often contain traces of iron and manganese. Treatment of this water typically involves removing these metals through different filters. Limestone is sometimes added to stabilize the pH-levels of the water. As a final step the water is passed through a UV-filter to kill any microorganisms before being sent out to consumers [6].

Surface Water

Groundwater in Sweden typically holds a higher quality than surface water, needing less treatment before the water is safe to drink. The treatment process for surface water is almost identical to the process of wastewater treatment. With the major differences being two additional steps after the treatment of the water. The first being the adding of CO_2 and limestone to stabilise the pH-levels of the water, while the second step involves passing the water through a UV-filter to neutralise any microorganisms.

2.3 SCADA

Since the start of the third industrial revolution and the introduction of computers to industry in the 1970s the importance of monitoring, visualizing and controlling has grown. Huge leaps in computer technology have introduced wireless and wired means of communications, such as MODBUS, OPC DA and OPC UA[7]. One result of these advancements is that it becomes manageable to collect and aggregate large sets of data. This in turn empowers users such as operators, engineers and maintenance personnel with more advanced analytical tools, which allows for reduced costs, preventing unnecessary repairs and better management of time and the plant in question [8].

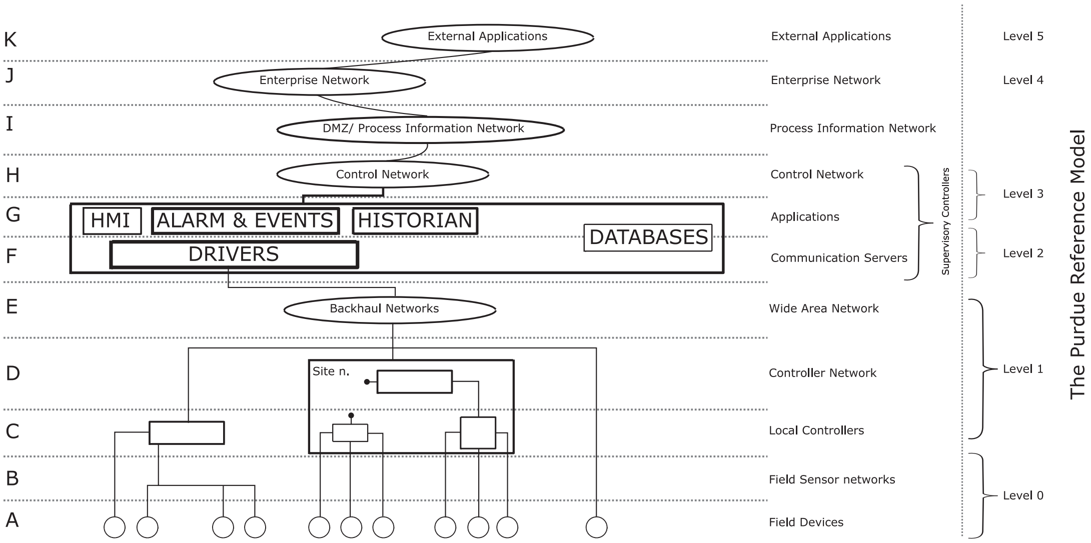


Figure 1: Enterprise System Model Architecture Diagram

The details of a SCADA system may vary depending on factors such as size of operations and type of process to monitor. One way to define where in the architecture of an enterprise a SCADA system can be found is through the Purdure Reference Model [9]. The Purdue Model can be used as a tool for modeling an enterprise in multiple layers, see figure 1. The ISA-112 standard uses the model to define the essential components for a SCADA system at level G and F in the figure [10].

Drivers

Drivers are what connects the SCADA and all systems above level 1 in The Purdue Reference Model to the plant and its devices. For WideQuick the most typical drivers would be those that enables WideQuick to communicate over an OPC or Modbus connection.

Historians and Databases

Historians and other types of databases are used to archive and store historical data from the process. The data collected in these databases have different use cases depending on who access the data. Users can for example use the database to:

- Store process data, events and alarms, operator actions, calculations and application results, etc.
- Storage of planned future data. The historian is sometimes used for planning and optimization functions.
- Definition of user calculation, for instance to detect a leak from in- and outflow or accumulated water usage over a period of time.
- Play back of process data using the normal process displays.
- Presentation of trends and reports.
- Programming of advanced statistical applications that use historical data to find changes in process characteristics.

In WideQuick any database that can connect with a ODBC connection is viable. However, relational databases, such as MySQL and MariaDB, are recommended and tested.

HMI

A Human Machine Interface or an HMI is a user interface that connects the operator, system integrator or engineer to a machine, process or device. This is described in greater detail in section 2.4.

Alarm and Events

Alarm and events help to focus the attention of the user towards critical operations, maintenance and faults. This is typically communicated by alarm lists and alerts in the SCADA application but also through e-mail and text messages to on-call personnel depending on the severity of the event or alarm.

2.3.1 Wastewater and Water Treatment Gains from SCADA

A SCADA system can be found in many different fields where water and wastewater treatment is one of them. Much of the functionality in a SCADA is the same independent of what field it is applied to, some functions that are used for water and wastewater treatment are:

- Chemical details (dosage, storage, preparations)
- Leakage (calculating in- and outflows during a stable and predictable time reference)
- Efficiency Key Performance Indicators
- Reduce waste
- Weather reports and their impact on the process, e.g. incoming flows
- Average flow during recent, past, or similar time period for previous year
- Ensure optimum operation mode in the event of damage
- Monitoring status of remote systems such as a small pump station

2.4 Human Machine Interface - HMI

A Human Machine Interface or an HMI is a user interface that connects the operator, system integrator or engineer to a machine, process or device in real-time. The user interface can be displayed on any screen but in an industrial setting often displayed on an HMI-panel or PC [11]. The HMI is typically used to used to:

- Display schematic and geographic views showing the real time state of the plant or system
- Events and Alarms
- Trends and Reports
- Tabular displays
- Control and configuration of field devices
- Maintenance and diagnostic views

A SCADA and an HMI are closely related as they are often referred to in a similar context. HMIs however do not collect data and store it in a database as a SCADA does but rather focus on conveying information to the user in an effective way alongside the SCADA.

The HMIs that can be encountered in industry today were first introduced in the 1980s, replacing "control walls" with electromagnetic and pneumatic instruments. These HMIs generally included low levels of abstraction in what information was displayed to the user and therefore could be considered cluttered and deprived of useful information. A solution to this problem is a set of best practises and frameworks called High Performance HMI [12].

2.4.1 High Performance HMI

The implementation of a high performance HMI is realized with the understanding of a four key-display hierarchy and a clearly defined HMI philosophy and style-guide [12].

The hierarchy can be split into:

- Level 1 - The Process Overview
- Level 2 - The Process Unit Control
- Level 3 - The Process Unit Details
- Level 4 - The Process Unit Support

Level 1 - The Process Overview
Level 1 is the first layer of the display hierarchy and is responsible of conveying an overall situational awareness of the system. Typically the overview would contain key process value indicators, major equipment status, alarms of high priority and indications of abnormal situations.

Level 2 - The Process Unit Control
Level 2 should include more detailed graphics, controls and information for a specific plant unit or sub system. In the design process level 2 should be created first and include Piping and Instrument Diagram (P&ID)-like graphics of the system, more in depth controllers and key performance indicators.

Level 3 - The Process Unit Details
Level 3 contains information on a specific object or piece of equipment, for example a pump. This would typically include trends of historical data, control signals and alarms specific to the device. The purpose of this level is mainly to access more in depth manipulation and troubleshooting.

Level 4 - The Process Unit Support
Level 4 should contain detailed information about the equipment such as ratings of a motor and help documentation for maintenance and worklogs.

2.5 Communication

In order for the SCADA system to communicate with the physical sensors/field units different means of communication are used. The most common ones are MODBUS and OPC, see figure 2. WideQuick supports both of these. The general idea behind both of these are the same but they work in fundamentally different ways.

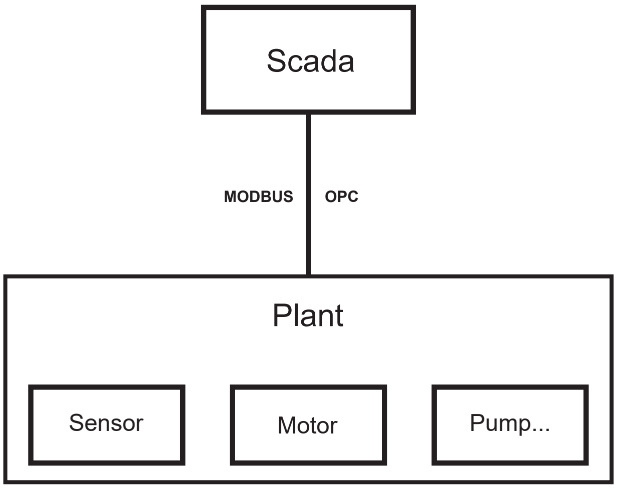


Figure 2: The structure of communication between the SCADA and the physical world

2.5.1 MODBUS

Modbus is the oldest of the two communication protocols. Developed in 1979 by Modicon (now a part of Schneider Electric). Modbus is a Client/Server protocol. In simple terms this means that the server, often a PLC does not send any data until the client, in this case the SCADA, requests it. In other words, there is no way for a PLC to send data to the SCADA if a sensor has a change in value. In practice this means that the SCADA will have to request data from all PLCs to look for a change in the data, even if the data might be unchanged since last time. An exception to this is OpenMbus, where the client is able to subscribe to the server and which then sends an update every time a value changes.

WideQuick supports two different communication protocols for Modbus. Both Modbus TCP and Modbus RTU. The two protocols are essentially the same, with the difference being on how the data is sent, see figure 3. Modbus serial works as the name suggests by using serial communication. In most applications this means using RS-232 or RS-485. Modbus TCP on the other hand uses the TCP/IP protocol.

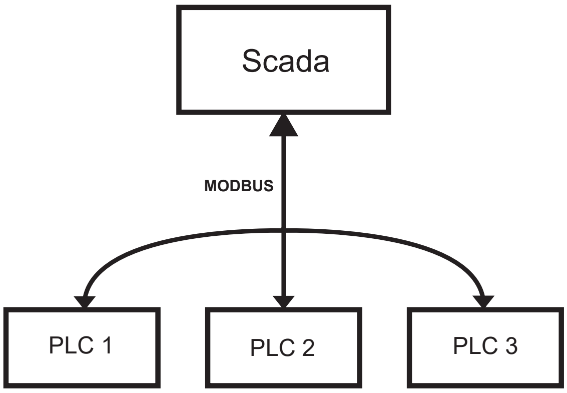


Figure 3: The structure of Modbus communication between the SCADA and the physical world

2.5.2 OPC DA/UA

OPC (Open Platform Communications), formerly **O**bject **L**inking and **E**mbedding for process control was developed in 1996. OPC was initially developed for Microsoft Windows, called OPC DA (Data Access). One problem that OPC tried to solve was: PLCs from different manufacturers used their own, brand specific communication protocol. This meant that each HMI/SCADA developer had to program the driver for each communication protocol and implement it into the SCADA. OPC solved this by placing an OPC server as the "middle man" for the communication, see figure 4 . This means that the OPC server would still need to be able to interpret all different communication protocols, but could then translate them all into a single, standard protocol, which then any client, in this case the SCADA system, could interpret. One advantage with OPC over MODBUS is that clients can subscribe to different tags. This means that the OPC server can send data to the clients only when a value, for instance a sensor measurement, has changed. One problem with OPC DA is that it is dependant on Component Object Model (COM) and Distributed COM (DCOM). COM still exists but is no longer officially supported by Windows. That is why OPC UA was developed. OPC **U**nified **A**rchitecture is as the name suggest a unified architecture, which in practice allows different types of operating systems to communicate using OPC.

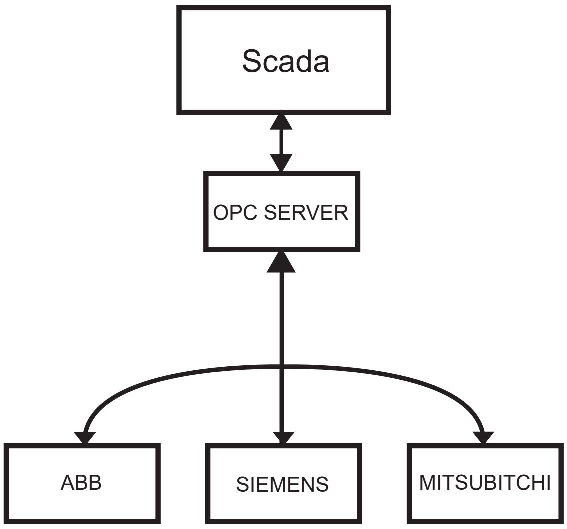


Figure 4: The structure of the communication between the OPC server and the physical world, as well as between the OPC server and the SCADA

3 Method

In this chapter a thorough explanation will follow, explaining the steps taken to develop the application. The work will be divided into four parts. First creating a list of requirements, developing the application, followed by commissioning example plants for demonstration purposes, and finally using the feedback from the demonstrations to update the list of requirements. See figure 5 for a more detailed overview.

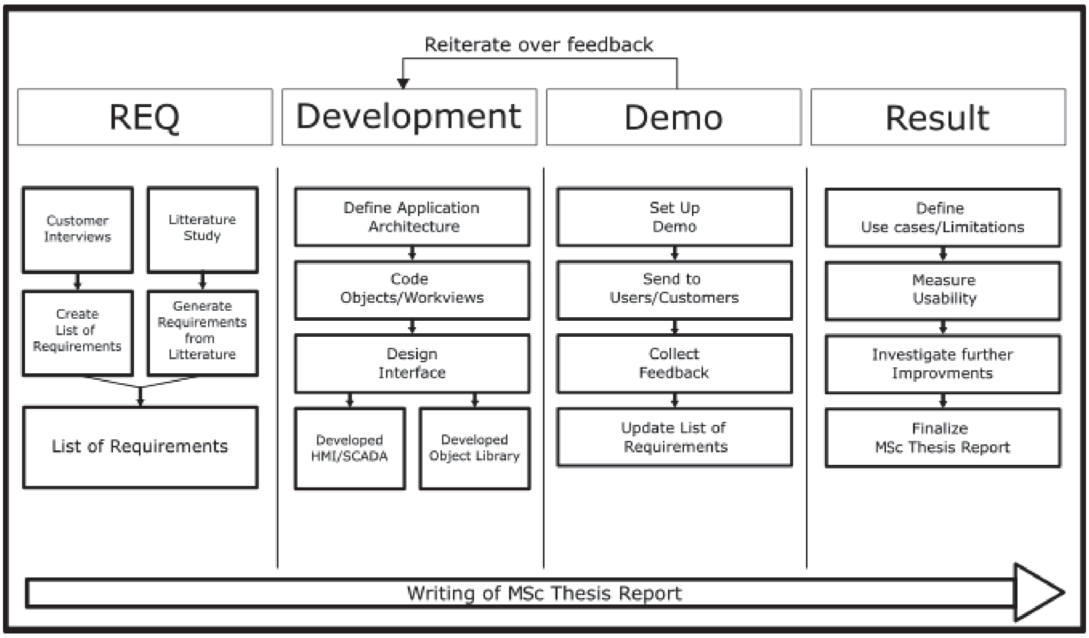


Figure 5: Visual aid for understanding the method

3.1 Preparatory Work/REQ

Prior to the development phase of the project preparatory work had to be completed. The goal of this work was to create a list of requirements, which would lay the foundation for the development of the product. The preparatory work was divided into three separate parts. Literature review, interviews and a list of requirements. The work process was also conducted in that order. Below follows a more detailed description of what each of the parts entailed.

3.1.1 Literature Review

A literature review was undertaken to give relevant theory to support the decisions made in the design of the application architecture and HMI design but also to get an overview of the processes in water and wastewater treatment to deepen the understanding of the needs of the industry.

Initially a brainstorming session was conducted to come up with relevant search words for the study resulting in a word cloud, depicted in Figure 6.

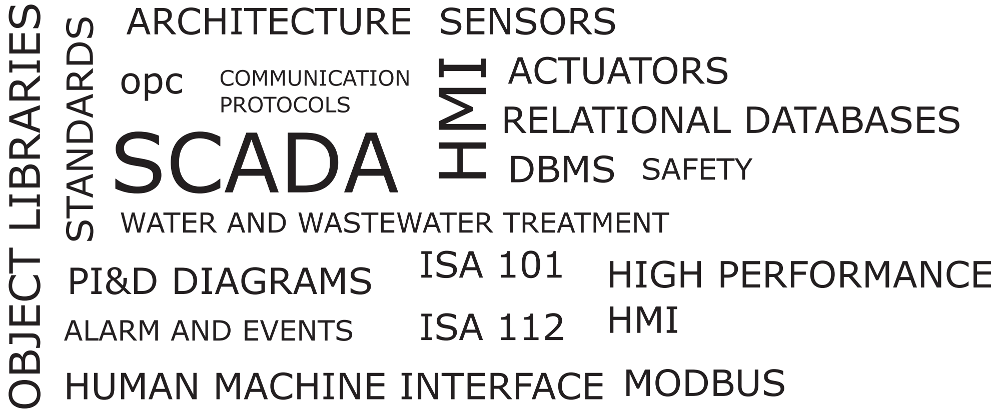


Figure 6: Search words used in the literature search. The size of the text highlights the relevance

The collecting of relevant texts was done with the help of databases such as LUBsearch and Google Scholar as well as the Lund University Library.

3.1.2 Interviews

Interviews were conducted in order to get an understanding of how a SCADA works for water and wastewater functions in real world applications. To assist with this Kentima referred four different customers in the water and wastewater industry to the authors. Since water and wastewater plants are protected objects a non-disclosure agreement, an NDA, had to be signed, which means that the customers can not be listed by name.

- **Customer A:** The biggest customer, responsible for several larger plants as well as many smaller ones such as pumping stations over a larger geographical area with a relatively high population density in multiple municipalities.
- **Customer B:** A consulting firm focusing on development of infrastructure with presence in Northern Europe, Africa and the Asia-Pacific region. The firm does not actively manage any plants on their own. Instead helps other companies and municipalities in setting up SCADA solutions for their sites.
- **Customer C:** An environmental resource company tasked with handling the water and wastewater treatment in a large but sparsely populated municipality.
- **Customer D:** A municipality with its own department of water and wastewater with a few treatment plants and pumping stations. The municipality is large to its geographical size but sparsely populated.

Before the interviews began, a script was written. The idea behind the script was to ensure that all interviewees would get asked the same questions since the interviews would only be held once per customer. Notes were taken summarizing the most important things said during each part. The interviews were divided into the following five parts:

- Introduction - Explaining who we are and why we are doing this.
- What are you using today - Which HMI/SCADA solution are you using today, what do you like? What can be improved?
- I/O list - What objects are going to be needed?
- Process design - What does the process look like? What does the system architecture look like?
- Standards - Are there any standards concerning the subject, tag structure, color coding and so on.

The full list of questions, i.e the script, can be found in the appendix.

3.1.3 List of Requirements

A list of requirements was composed from the interviews and literature review. It became apparent during the interviews that all the customers had widely different experiences using SCADA- and HMI-solutions. This meant that they had different wishes for what they wanted, both in terms of functionality and design.

Using the interview in conjuncture with the literature review allowed for the creation of a list of requirements. A summarized list of the most important findings can be found below:

Setup

- Short time to market.
- Quick and easy setup - Particularly simplicity in connecting all associated tags to an object.
- Support for both OPC and Modbus communication.
- Easy to scale from small to large systems.
- Ability to use the application either as an HMI or a SCADA system utilizing a local/remote database.

Usage

- Easy user handling - Both adding/removing users and privileges while the SCADA is running. Especially important for those with temporary workers and/or consultants who only need access for a limited amount of time.
- Alarm handling - Possibility of sending out alarms, both via text message and e-mail. But also configure where alarms will be sent depending on who is working by using a scheduling function.

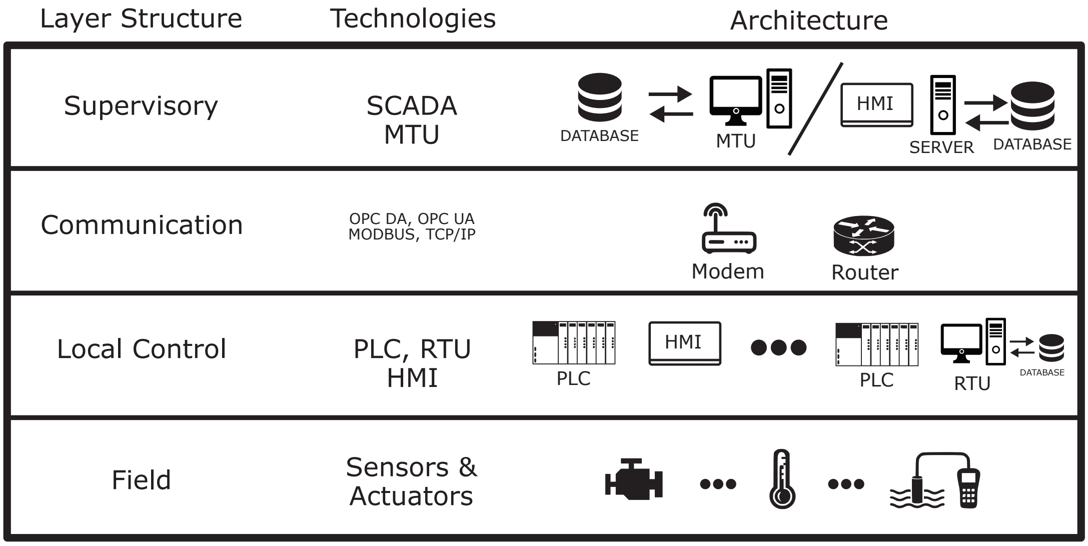
- Machine cards - A page where technicians can fill in any service completed as well as keeping track of calibration and maintenance dates.
- Animations on objects - Some operators are color blind or come from a different cultural background, so using the color green for ON and red for OFF is not a good idea. Instead focusing on animations, so for instance when a valve is closed it is turned perpendicular to the pipe, mimicking the real life scenario, so that it is obvious that it is closed.
- A hierarchy of what data to show when, so to only provide information when needed and not clutter the screen with data that gives little or no information.

3.2 Development

In this section the development of the application and choices made will be explained. It introduces the reader to some of the choices that were made that limit but also enables the user with different configurations of tags and architectures.

3.2.1 Application Architecture

An application architecture was designed with the interviews and literature in mind. Customers of Kentima wanted to be able to scale the application up or down according to the size of their operation. By letting the application act as both a SCADA and HMI depending on what level of the architecture you place it and to where it stores its data this becomes possible. See figure 7.



Layer Structure	Technologies	Architecture
Supervisory	SCADA MTU	DATABASE MTU / HMI SERVER DATABASE
Communication	OPC DA, OPC UA MODBUS, TCP/IP	Modem Router
Local Control	PLC, RTU HMI	PLC HMI ●●● PLC RTU DATABASE
Field	Sensors & Actuators	●●● ●●●

Figure 7: WideQuick Water and Wastewater application architecture

This was realized by allowing the application to run as a master terminal unit, an MTU, either on a machine or server that is connected to a database. Data from downstream remote terminal units, RTUs, and their respective field layers can then be communicated and collected in the database at the supervisory layer. The application can also be run as an RTU and collect data locally to its own database. This could prove helpful for an organization that for example manages multiple water and wastewater treatment systems in more than one municipality. They could then gain a level of abstraction and connect another WideQuick Water and Wastewater application to track data from all the subsystems.

A relational database was chosen as the database type of choice for this application. It was done so because relational databases are what WideQuick is tested against but also because the type of data gathered typically are data from field units over time. To create new tables in the database the application relies on a native database connection function in WideQuick Designer.

By creating a new "Database Connection" in the designer. A tag can then be added to the database connection by inputting the name of the tag. A part of the tag name can also be added where all tag with the same name up to that point will be added to the database if it is followed by an asterisk.

3.2.2 Tags, Tag Structure and Suffixes

Tags are in simple terms the name of different sensor measurements, machine settings etc. In practice this means that every different signal that can be sent to or from any machine or sensor is a tag.

Take a pump as an example:

1. Power consumption (kW)
2. Pump flow (m^3/h)
3. Setpoint for pump flow (m^3/h)
4. Is active (true/false)
5. Has any active alarms (true/false)
6. Manual mode active (true/false)

As can be been seen a single pump can have several tags. It is not uncommon to see even more tags for a single machine. In reality this means that a relatively small pump station can have over 100 tags in total. In order to organize all tags and to be able to connect them to the correct object in the SCADA, a specific tag structure with accompanying suffixes was developed which will be described in more detail below.

Tag Structure

Due to the difference in the protocols between OPC and MODBUS the different tags are addressed differently. However, WideQuick allows the user to set any name as a "WideQuick name" to these tags. This name is then what will be displayed in the program, allowing MODBUS and OPC tags to be used indifferently.

A dot notation "." marks the division of the tag structure, see figure 8. An exception to this was made between layer three and four where the layers are divided by an underscore "_" instead. For this project it was decided that a five-layer structure should be the standard. It offers a balance between organisation and order without creating unnecessary complexity. The layers are called: Connection, Device, System, Object and Suffix. The user can choose any name for these layers with the only restriction being that they all must exist. So following the top path in figure 8 would yield a total of 3 tags:

1. Skåne.Lund.AS01_Pump_IO

2. Skåne.Lund.AS01_Pump_MV
3. Skåne.Lund.AS01_Pump_MB

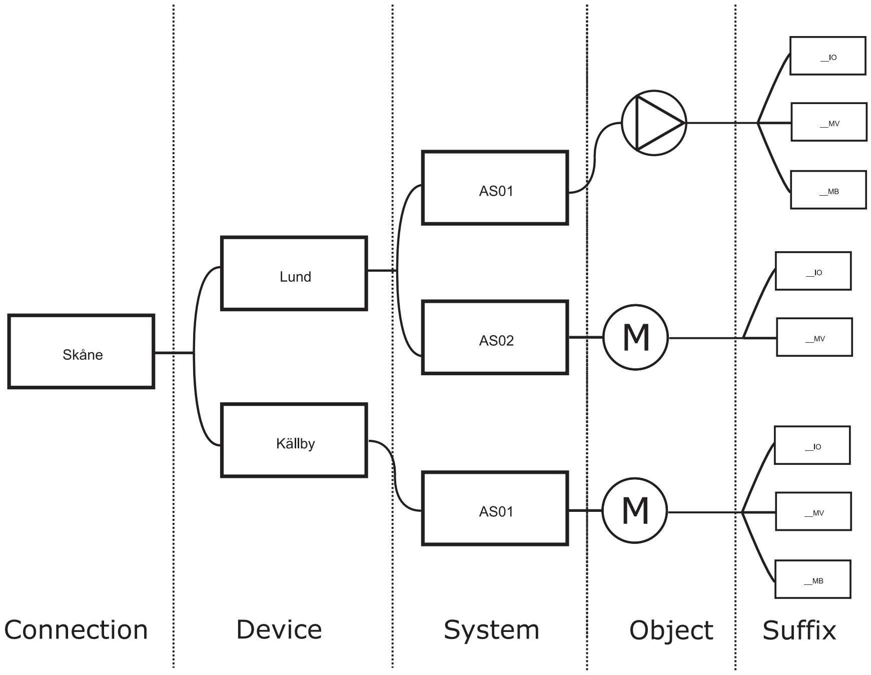


Figure 8: The tag structure used in the WideQuick Water and Wastewater application

Suffixes
Suffixes have a special meaning compared to the other layers. The first four layers are used to connect specific tags to an object and can be named arbitrarily. The fifth and final suffix layer however can not be named arbitrarily. The suffix defines what type of signal the tag is. For instance "_MV" means that it is a process- or a measurement value. Another example is "_IO" which means "Is open". These suffixes are however fully customizable, and the user can both remove and add suffixes.

3.2.3 Icons and Support Objects

As stated by several customers during the interviews, there is no standard solution for water and wastewater treatment plants. Because of this, it will be impossible to create standard objects applicable to any organization and plant. To solve this, it was decided to separate the UI-elements, such as the icons/symbols and support objects from the functionality brought by the "smart object". This in turn allows the user to create completely unique and complex objects, see figure 9

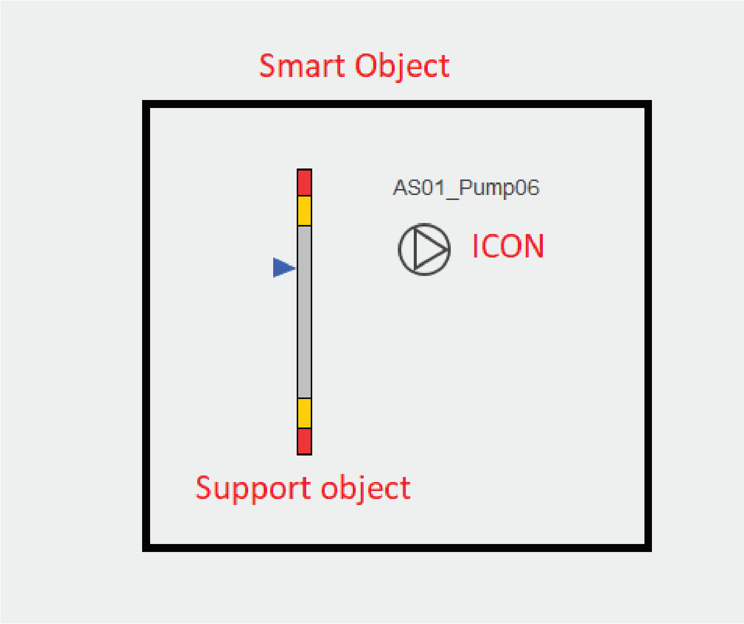


Figure 9: A smart object containing an Icon and a support Object. The text in red has been added by the authors clarify the different components

Icons

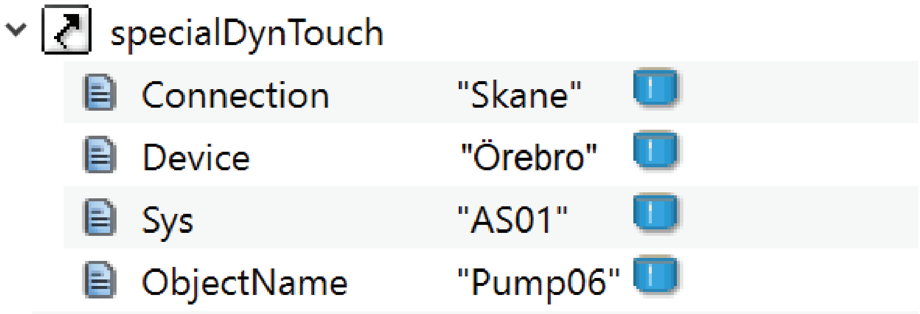
As different system integrators prefer different icons to represent their field units it was chosen that icons representing an object should not limit the object it is representing. For instance, an object can have the icon of a valve, but still be connected to the tags of a pump. The only downside of using the "wrong" icon is that some of the icons have simple graphical logic connected to them. They turn yellow or red when they have an active warning or alarm tag, this is a redundancy on top of the alarm symbols that becomes visible . Icons also rotate when they are on/off or open/closed. But this does not affect the overall functionality.

Support Objects

In order to assist the users in creating their own objects and to display the correct or most important information a collection of support objects were created. Support objects are used to display the value of a tag by allowing the user to select a suffix that should be displayed and connect it to a smart object. This in turn means that the value of a tag will be displayed right next to the icon in the user interface for the object without forcing the user to open the popup for the object.

3.2.4 Connecting Tags to Objects

Once all tags are named according to the predefined structure, the next step is to connect them to objects in the workviews. This is done with a component from the object library called dynTouch. The settings for the dynTouch can be seen in figure 10. The program will then use this information to find all tags belonging to this specific object.



specialDynTouch

Connection	"Skane"
Device	"Örebro"
Sys	"AS01"
ObjectName	"Pump06"

Figure 10: The settings for the dynTouch

3.2.5 Smart Objects

Smart Objects are the result of connecting tags to an object using the object "dynTouch". In essence they way of visualizing the data from all connected tags. They do this in several layers according to the high performance HMI structure.

The Process Unit Control
The top-most layer where the specific object is displayed is in the process unit control. In this layer just the most important information is displayed. Such as the icon for the object, as well as any support objects that may have been connected to the object. This is to give a quick overview of the most important parameters, such as important measurement values and alarm status. All without overwhelming the user with information.

The Process Unit Details & Support
Clicking on an object starts an object-specific popup with several views. These are explained in greater detail in the chapter 4 under "Common Popups". But some examples includes "control" and "history". The control page acts as a two-way communication for the user and the object. The user can set values to tags, for instance enabling manual mode, changing the set-point for a machine or starting/stopping a machine. In turn the object can display status for different control parameters, such as on/off indicator or auto/manual mode.

The history popup on the other hand is a way for the user to access the object specific tags in the database. The user only has to select which tags they want to view from the historic data and the program will automatically formulate an SQL-Query and send it to the database. The returned data will then be displayed.

3.2.6 Documentation

As mentioned by several customers during the interviews the manuals provided for other WideQuick project have been quite difficult to use. This is due to several reasons. Taking the manual for WideQuick BMS as an example:

- The manual is a single PDF document with over 90 pages, making it quite difficult to find the information that is needed
- The only way to search for a specific topic in the manual is by using the table of contents at the beginning or by using search (CTRL + F), however this is not 100% accurate and finding the correct information might be difficult
- Since the manual is in the form of a PDF that is sent out with each release, there is no way to update the manual if any bugs are found or to add any information regarding a subject

To solve this it was decided to move the manual to a web-page instead. This would ensure that the manual could be updated with the latest information. It would also allow for a FAQ-section with frequently asked questions. Furthermore, version handling for the manual would also be simplified using smart functions.

MkDocs was chosen for this purpose. MkDocs is static site generator built for generating project documentation. MkDocs is written in a language called MarkDown, which is very similar to plain text. A plug-in called Material for MkDocs was also added [13]. This allowed both the ability to search and even more extensive customization. There are also converters between JavaDoc[1] and MkDocs. This means that all the documentation for the script in the project can easily be converted and pasted into the documentation. The manual contains both explanations of all scripts, objects and workviews. But also user guides describing how to create custom objects and workviews. This will be updated in the future to include a FAQ-section and version handling. More on this can be found in chapter 6.

[1] JavaDoc is a documentation generator in Java/Javascript. It creates a documentation using the input parameters and expected output

3.3 Demo

An initial demonstration was held at Kentima with the developers of the WideQuick software, the sales representatives as well as management. The idea behind the demonstration was to showcase the application and its functionality. This was also done to get the approval to demonstrate the application to the customers, a showing of an incomplete or non-working application to the customers could potentially hurt the relationship with said customers.

After getting approval from the company, the planing for the customer demonstration began. Since water and wastewater plants are protected sites anonymity for the customers is key. Therefore, instead of booking one meeting with all customers and showing off the application, several meetings were booked.

3.3.1 Example Plant

For this project two example plants were developed. One water treatment plant called: Källby-Skattegård and one wastewater treatment plant called: Videdal-Intake. The idea behind the demo plants was twofold. Firstly, they were used as a development tool to test new functions and to test how the different functions would work together. Secondly, and perhaps even more importantly is to have a working plant to show the customers. This will give the customers the ability to immediately compare the application to the solutions that they are using today.

In order to make the example plants more realistic, all objects were given simulated tags with the help of KEPServerEX. This means that they will have measurement values that change over time.

3.3.2 Customer Demonstration

The customer demonstrations were divided into the following five parts:

1. The demonstrations started with a Power Point presentation. The presentation consisted of a road-map, explaining what had been developed, what the next steps will be and what the final goal of the project was. The summarized list of requirements was also presented, with the intent of trying to explain some of the development decisions that had been made and also to explain the priority order for upcoming functionality.

2. Thereafter, it was time to showcase the application itself. Since all the customers are spread out throughout the country an online demonstration was held. The demonstration started off by showing the main map view, as well as a dashboard view containing KPIs for the two example plants. Next up were the two example plants, and finally showing of the auto-generated popup for objects. The second part of the demonstration was concluded by asking the customers for their first impressions of the software.

3. Next up was showcasing the simplicity of creating a process. This included creating custom plants (workviews) as well as custom objects. Simplicity in the creative process is vital to ensure a short time to market. Which was one of the requirements.

4. One of the most important requirements was flexibility. Since there is no standard when it comes to water and wastewater SCADA, the need for flexibility and custom configuration is essential. Therefore the next part of the demonstration was to showcase the many customization options available. This includes: defining custom colors for the project, how to create custom suffixes and how to work with users and privileges.

5. The final part of the demonstration was showcasing the documentation. Good documentation is vital to ensure that users know how to use the project as well as knowing its limitations.

3.3.3 Customer Feeback

After each part of the demonstrations, a short break was held. Both to answer any questions and to collect any feedback that the customers had. This was done between each part instead of after the demonstration as any feedback they might have had during the first parts could easily be forgotten during the demonstration.

The feedback that was given during the demonstration was sorted into two categories. The first category containing feedback on the parts that had been developed. The second part was more of a wish list for things that the customers wanted to see in the final product.

A common complaint was that it was quite hard to give detailed feedback without using the software themselves. After relaying this to the company supervisors a special agreement was reached between a few selected customers and Kentima to allow them early access to the project.

3.3.4 Updated List of Requirements

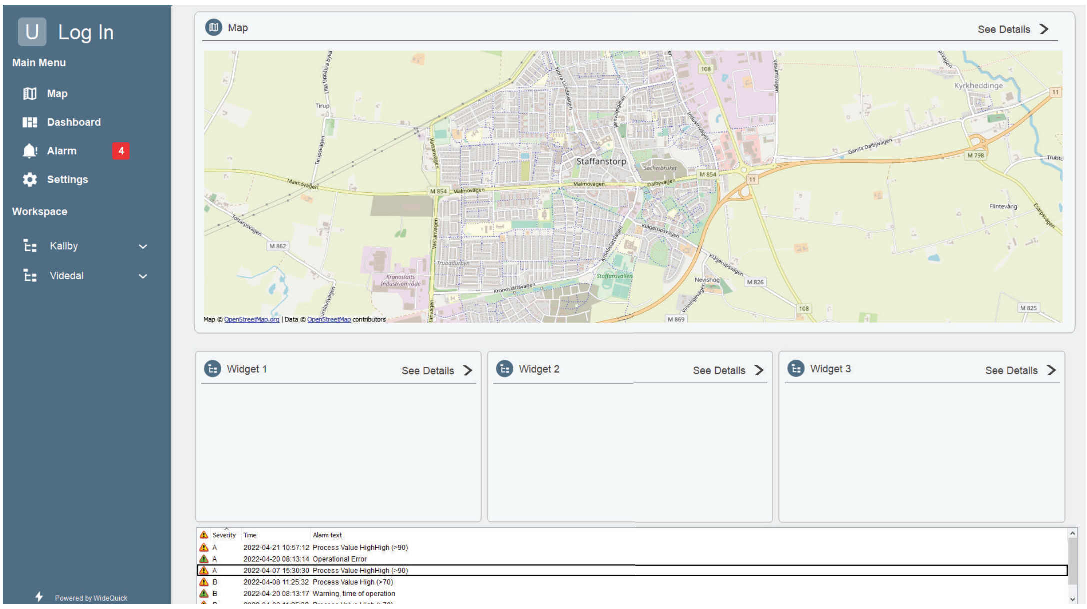
Using the feedback from both the demonstrations but also from the special early-release access an updated list of requirements could be created. The list contained parts that needed to be improved but also functionality not yet developed.

4 Results

In this cahpter the result of the developed application is presented. The results are divided into three layers where layer one describes the overview of all systems and correlates to level one in the high performance HMI as described section 2.4.1. Layer two correlates to level two in the high performance HMI hierarchy and speaks of sub systems and objects in the workviews. Finally, layer three presents the details of a specific object and correlates to levels three and four of the hierarchy. Results for documentation and customer feedback are also presented.

4.1 Layer 1. The Overview

The overview of the application is presented in the first layer. This layer gives the user an overview of all systems that are connected to the SCADA. The layer consists of four workviews and a navigation bar. The workviews provide the user with different information to supervise their enterprise's control systems.



Severity	Time	Alarm text
A	2022-04-21 10:57:12	Process Value HighHigh (>90)
A	2022-04-20 08:13:14	Operational Error
A	2022-04-07 15:30:30	Process Value HighHigh (>90)
B	2022-04-08 11:25:32	Process Value High (>70)
B	2022-04-20 08:13:17	Warning, time of operation

Figure 11: Example of a view in hierarchy level 1, the map workview

4.1.1 Navigation

The navigation bar, as seen to the left in Figure 11, enables the user to log in and traverse the application to the sub systems and their different views. The navigation bar will be visible on all the levels of the application so to enable the user access to the overview even when viewing a sub system of the SCADA.

Function	**Description**
Log in	Lets the user log in with a username and password. Each user has different privileges which in turn enables functionality depending on which privilege is granted
Map	Link to an geographical overview
Dashboard	Link to a overview of important KPIs
Alarms	Link to an alarm overview. Contains alarms from all systems connected to the SCADA
Settings	Settings which affect the whole SCADA application
Workspace	Link to subsystems defined in workspace catalog, see section 4.2.3
Sum Alarm	When there are alarms active a sum alarm becomes visible by the alarm button in the navigation. This sum alarm tells the operator how many alarms that are currently active and unacknowledged.

Table 1: List of functionality accessible from the navigation bar

Users can be added during runtime as long as the user currently logged in have the correct privilege. In addition to this, users and privileges can also be inherited through the active directory on a Windows machine or domain. Though this has to be activated through the developer tool WideQuick Designer. Adding another user in runtime can be done in the settings workview.

4.1.2 Dashboard

The Dashboard provides the user with an interface where the system integrator can create a number of sub systems of their choosing. In these sub systems the system integrator can use smart objects connected to tags to display KPIs, alarms and events or historical data to get a data driven overview.

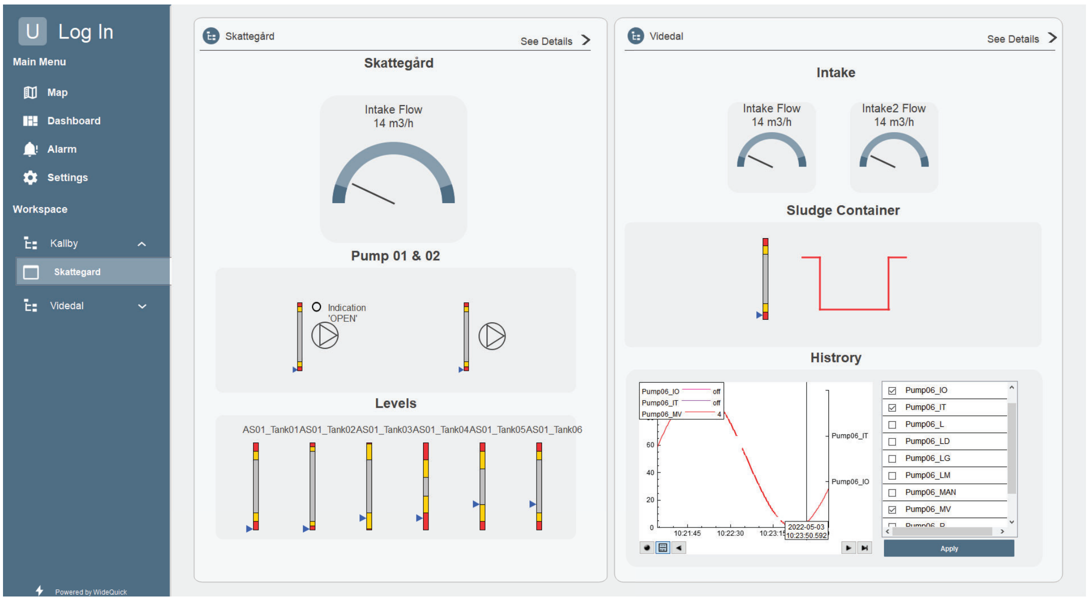


Figure 12: A snapshot of an example Dashboard workview

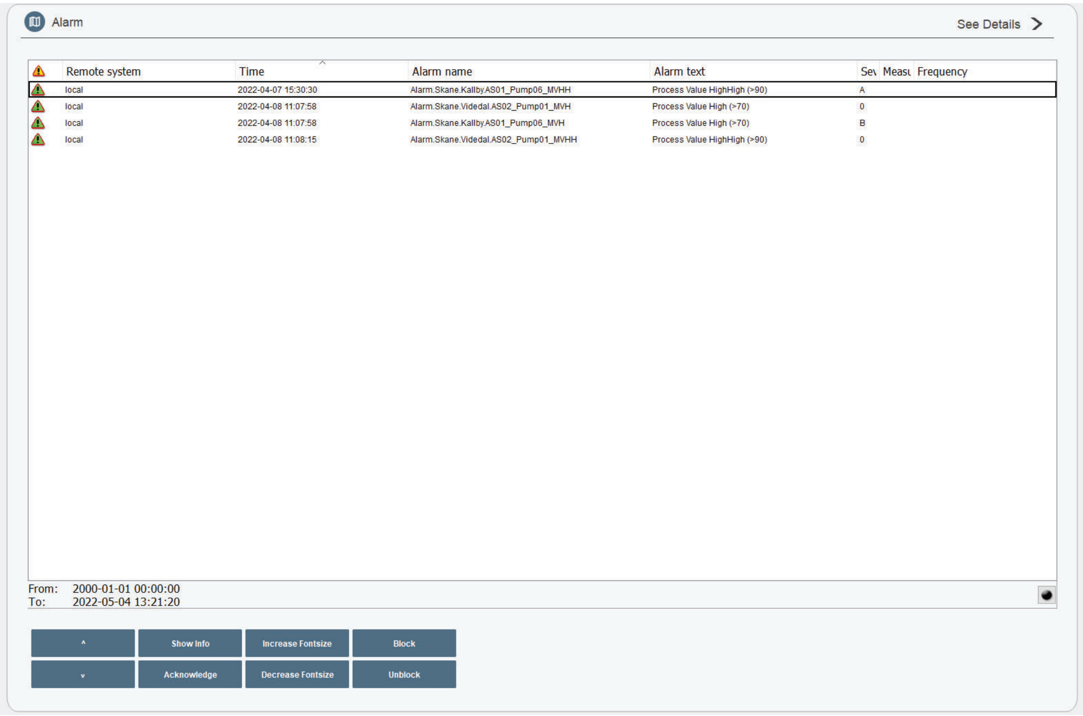
An example view of a dashboard is given in figure 12. The dashboard contains two sub systems and highlights the intake flows in cubic meters per hour into the systems. It also displays levels in different vessels throughout the systems. This helps the user get a more condensed overview of levels than they would in a workview designed to look more like a P&ID diagram. In addition to this the Videdal system also has a history view where the different tags connected to a pump are displayed.

Function	**Description**
Sub Systems boxes	A graphical container to differentiate between systems
Smart objects	Objects connected to tags to provide information about a certain machine or process. See section 3.2.5
Historical Data	Displays historical data from a database to an object or workview.
Alarm and Events	Lists alarms of a certain severity for a sub system

Table 2: Example of functionality that can be displayed in the Dashboard workview

4.1.3 Alarms

The alarm view provides the user with an overview of active and archived alarms throughout all connected systems. An important use for this can be when maintenance personnel wants to get an overview of problematic sensors or when a process engineer wants to analyse problem areas in the process.



⚠	Remote system	Time	Alarm name	Alarm text	Sev	Measu	Frequency
⚠	local	2022-04-07 15:30:30	Alarm.Skane.Kallby.AS01_Pump06_MVHH	Process Value HighHigh (>90)	A		
⚠	local	2022-04-08 11:07:58	Alarm.Skane.Videdal.AS02_Pump01_MVH	Process Value High (>70)	0		
⚠	local	2022-04-08 11:07:58	Alarm.Skane.Kallby.AS01_Pump06_MVH	Process Value High (>70)	B		
⚠	local	2022-04-08 11:08:15	Alarm.Skane.Videdal.AS02_Pump01_MVHH	Process Value HighHigh (>90)	0		

From: 2000-01-01 00:00:00
To: 2022-05-04 13:21:20

Figure 13: The alarm view

Function	Description
Acknowledge	Acknowledges the alarm, removing it from the count of active alarms
Block/Unblock	Blocks/unblocks an alarm. A blocked alarm will never trigger and does not count up the number of active alarms
Traverse list	Traverses the alarms in the list by pressing up or down arrow
Change font size	Increase/Decrease the font size of the alarms in the alarm list

Table 3: List of functionality accessible on the alarm page

4.1.4 Settings

The settings view allows the user to control certain aspects of the application while it is running. Some examples are switching to dark theme, change the font size and show/hide the borders of the dynTouch object. A full list of settings available to the user can be found in table 4.

Function	**Description**
Language	Enables the user to change the language of the application. **Note:** The application is developed in English. However, a translation can be provided by the system integrator.
Unit	Allows the user to switch between different unit systems, such as imperial and SI.
Time	Allows setting the system time.
Schedule	**Under development:** Enables the creation of schedules. For instance creating different groups for the day and night shift. This in turn could be used to only send alarms to whoever is working at a specific time.
Remote Systems	**Under development:** Allows the configuration of remote systems.
Aliases	Enables the creation of aliases, which is a collection of e-mail addresses and phone numbers. In practice this means that the system integrator can create groups of people under one name, for instance "night shift". Then using schedules to send alarms to the entire group.
Light/Dark theme	Allows switching between a light and dark theme of the application.
dynTouch border	Hides/shows the borders of the dynTocuh (the clickable area of an object).
Font size	Changes the font size of support objects.
Edit users	Handles the creation/deletion of users, adding/removing privileges and changing of password.

Table 4: List of functionality accessible from the Settings view

4.2 Layer 2. The Sub Systems and Unit Control

The unit control layer contains several different components. They are for instance UI-elements, such as icons and symbols and support object which display information about tags. These in turn make up smart objects where smart object in turn make up subsystems and workviews such as the demo plants.

4.2.1 Icons

Below follows a collection of P&ID icons that were created to represent objects. The icon themselves are not required to create an object. However, they do have some simple logic connected which enables them to provide additional information. For instance they will turn yellow or red if they have an active warning or alarm tag. A few selected icons have additional logic, such as the valve. The valve will turn 90 degrees depending on if it is open or closed. This means that the valve will turn perpendicular to the pipe it is connected to, meaning that an operator can quickly see that there is no flow. See table 5 for a selection of icons.

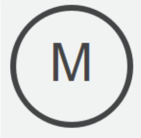
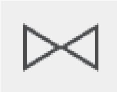
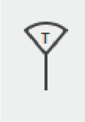
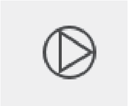
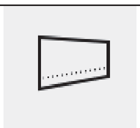
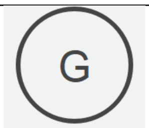
Name	**Symbol**	**Name**	**Symbol**
Motor		Valve	
Sensor		Pump	
Compressor		Generator	

Table 5: A collection of icons (UI-elements)

4.2.2 Support Objects

Support objects are used to relay information about an object or a part of a processes. Two different categories of support objects have been developed. First are those that only present raw data. For instance the current energy usage of a pump. The second category of support object are those that represent data in relation to min/max levels in accordance to high performance HMI principles. An example of this is a gauge. See table 6 for more information.

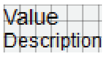
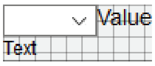
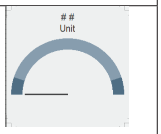
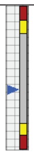
Name	Description	Image
Static Value	Displays the value of a tag defined during the design process and can not be changed while the application is running, hence static.	
Combo Value	Displays the value of a tag. A combo box allows the user to change which tag is tracked while the application is running	
Gauge	Display a value in relation to min and max levels	
Analog Value Display	Displays a value in relation to min and max levels, with both warning and alarm levels	

Table 6: Four different support objects. The first two only display raw data while the last two display data in relation to min and max

4.2.3 Sub Systems

Sub systems are a collection of workviews comprised of objects. Each such workview represents a process or part of a process. Examples of this could be a pumping station or the intake side of a treatment plant. The two demo plants developed for this project are sub systems. An example can be seen in Figure 14. Every object in the view is clickable. Clicking on an object opens layer 3. For now it is not possible to put another sub system into an already existing sub system and have it being displayed in the navigation bar.

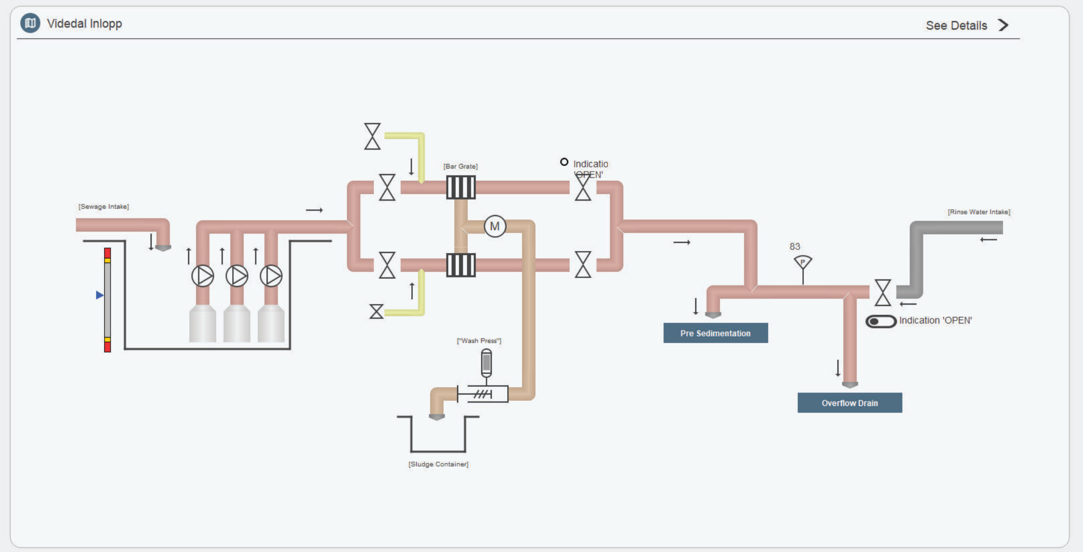
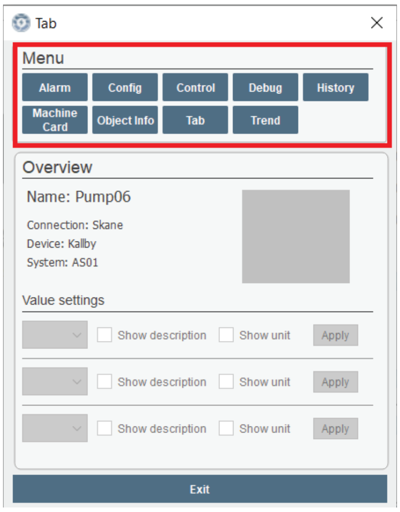


Figure 14: A process overview of "Videdal - Inlopp"

4.3 Layer 3. The Process Unit Details & Support

The third and forth layers are object specific. They are responsible for control and overview of an object. Clicking on an object brings up a popup with several tabs. See Figure 15, the different tabs are highlighted in red. Depending on user privilege, different tabs will be available. These tabs are referred to as "common popups" in this report, see table 7.



Name: Pump06

Connection: Skane
Device: Kalby
System: AS01

□ Show description □ Show unit Apply

□ Show description □ Show unit Apply

□ Show description □ Show unit Apply

Figure 15: The popup that is shown when clicking on an object

4.3.1 Common Popups

Workview	**Description**
Alarm	Displaying all user defined alarms specific to the object that opened the popup.
Config	Allows the user to set requirements on which user privilege is required to open a specific popup. Also allows setting requirements for which suffix a certain object should have to be able to show a certain popup.
Control	Displays all tags connected to the object which have a suffix defined under "control" and "control status" suffix groups. All "control" suffixes can be manipulated by the user while the "control status" suffixes only display their values.
Debug	The debug popup is intended for troubleshooting and not for day to day operations. The popup displays all tags connected to the object and allows the user to manipulate every single one, including starting alarms.
History	Displays all tags connected to the object. The user can then select which tags they want historic data from, the program will then automatically search through the connected databases in search of historic data for the specific tag. If more than one database connection contains a tag the one with the highest data point resolution is chosen.
Machine card	**UNDER DEVELOPMENT:** A view intended for service technicians. Allows the user to record any service they might have performed, as well as setting calibration dates for the specific object.
Object Info	Displays all tags connected to the object. Along with a short description for the suffix, the type of signal (boolean, number) as well as the unit.
Tab	The starting view when clicking on an object. Shows basic information about the object and its tags, such as connection, device and system.
Trend	Displays all tags connected to the object. The user can then select which tags to track which will be displayed just like in the history view. With the difference being that the trend view does not show historical data. Instead starts the tracking when the user activates the tag.

Table 7: The different common popups available

4.4 Customer Feedback

Here follows a list of feedback that the customers had:

- More than once the customer had difficulties remembering to add quotation marks around a property of string type.
- A customer requested easier access to tag layers Device, System and Connection as it become tiresome to add these for many object at a time.
- When using the support object "AnalogValueDisplay" a customer stated that they wanted to set alarms on raw values rather than percentages of the full range.
- One customer requested more advanced data analysis and graphics to present it. For example evaluating leaks by calculating inflow versus outflow a certain time of day when the flow is generally stable.
- A customer requested a larger variety of objects in the map view. For example to measure flow through water mains to better supervise loads in the system directly in the map.
- 3D-Pipe graphics are unnecessary. A line would portray just as much information and not clutter the view as much.
- A deeper max depth of directories in workspace would be appreciated as a systems might contain even more systems within itself.
- A status view for communication was requested. This would make it easier to overlook when communication fails.
- Graphics to display an average value on a tag for a relative time horizon. For example average inflow into a wastewater treatment plant over the last two weeks.
- A customer stated: "Not too many clicks to access the desired information yet also not too few as to clutter the whole project".
- A customer stated "It is easy to set up unique objects that represent the equipment that we have on hand in our plants. Easier than in the software we use today."
- It was conveyed that a customer would like a smarter history view. The history viewer should be able to display data from more than one database connection simultaneously depending on how time horizon as well as the resolution of the signal.

4.5 Documentation

The documentation is currently hosted using Github Pages[14]. This is a temporary solution to demonstrate the efficacy and simplicity of the page. The documentation is likely to be hosted on the Kentima website in the future. The landing page contains a short explanation of what can be found in the documentation as well as some common acronyms and abbreviations. Shortcuts for the different categories are also available.

4.5.1 Getting started

The getting started page contains a guide for a first time setup. It explains the steps a user has to take to set up a plant using WideQuick Water and Wastewater. It also contains links to other guides explaining the different steps in more detail.

4.5.2 Data Store

Under Data Store, three sub-categories are listed. These are: Internal variables, Alarms and Tag structure. Internal variables list all internal variables as well as an explanation of what they are and their purpose. Alarms, on the other hand, explain how alarms work, how to set them up and how to name them to ensure proper functionality. Lastly, tag structure explains how the tag structure is set up for this project, how to import both Modbus and OPC tags and how they work.

4.5.3 Object library

The object library contains all things to do with the creation of objects, such as a list of symbols/icons and support objects. Each support object also has a short description, explaining its functionality. Also found on the object library page are guides explaining how the users create their own smart object.

4.5.4 Scripts

Listed under script are all the scripts defined in the project. This includes all functions, a short explanation of the functionality, their input parameters as well as the expected output. Since all of this is already defined in the script files using JavaDoc, a converter between JavaDoc and MarkDown was used to migrate the function description into the documentation.

4.5.5 Workviews

Under workviews are guides explaining how to create new workviews, for instance a new plant. This is supplemented with an explanation on how the automatic navigation bar works, which automatically creates a navigation link to the newly created workview. Also listed under workviews are the common popups. Each popup has its own page, describing what information can be found on that specific page, any requirements for the page as well as a picture of the popup.

5 Discussion & Conclusions

This project was started on behalf of Kentima. They had requested an application for the water and wastewater industry, like the BMS application. The project was started with a literature review. The reason for this was that we wanted the latest academic knowledge on both SCADA in general and to get a solid understanding of SCADA in water and wastewater treatment. This would give a "best practice" to keep in mind while designing the application.

However, knowing that even if there is best practice or a recommended standard it is not always followed. A standard is really only a "standard" if it is commonly used in the industry. Therefore the interviews proved vital. The interviewees have experience using SCADA solutions for the water and wastewater industry and therefore listening to how they use the product is weighed more heavily than the literature as they are the intended users and customers.

5.1 Layer 1. The Process Overview

The top layer, the process overview, is meant to give the user a quick overview of the most important parameters and indicators from several processes in a single view. We tried to achieve this in a few different ways. In the application this was achieved by adhering to a high performance HMI style guide and hierarchy as far as possible. However, when the application is applied to a small pump station and used more like an HMI rather than a SCADA, this layer can be considered to be excessive. As explained in section 3.2.1 there is nothing in the architecture that limits the user from this kind of use but considering how the application is built today there is no HMI to match this use case.

From the customer feedback it can be concluded that more analytical and system-wide management tools and objects are desired. This is not surprising as much of these tools have been deemed outside the scope of this project and will therefore be discussed in chapter 6. Feedback such as *"Not too many clicks to access the desired information yet also not too few as to clutter the whole project"* prove that the hierarchical structure provided seems to work even though it deviates from the high performance HMI levels, considering only three layers are used instead of four as explained in section 2.4.1.

An HMI was designed for the workview Map along with all other workviews. Initially the development of functionality was also intended to be executed, however, as functionality of this view grew with customer requests and the complexity of said functions it was decided to postpone this work until adequate time can be allocated to finalize it.

A communication overview was requested in the feedback by several customers. This could prove especially useful when trying to troubleshoot the project as communication with each PLC could be monitored. The authors feel that layer 1 is the correct layer to place this overview as it is part of the overall system and not a specific subsystem. A similar view has been developed previously in WideQuick for another client. However, that solution required .NET plugins which means that functionality is only guaranteed on systems running Windows operating system.

5.2 Layer 2. The Sub Systems and Unit Control

Most of the feedback from the customers concerned the second layer of the application. One possible explanation for this is that a lot of time is spent in the level by the system integrator. This is especially true during the initial commissioning of a plant in the application. It is during this time that all the objects are created, connected to tags and supplemented with support objects. It is also in this level that one of the biggest advantages with the application can be seen. This being the very simple setup, which in turn drastically reduces the time to market compared to other products.

Customer feedback on UI-element varied. Some customers liked having more 3D-looking components and some would rather stick to a view that looks more like a PI&D-diagram. To minimize the workload of having to create symbols for both types a decision has to be made together with Kentima on what style to pursue. The benefits of having PI&D looking UI components is that it is more standardized and it clutters the screen less. One drawback of PI&D UI components is that it can be considered difficult to recognize what symbol represents what device.

The configuration of tags have been considered cumbersome by some of the customers. The reason given was adding many objects requires the user to input all the layers of the tag structure of an object. In the WideQuick BMS application this is partially solved by letting objects in a certain workview inherit part of its tag structure from the workview properties. This is useful when many of the layers are the same for multiple objects in a workview. An example for this would be a process where many of the field devices are controlled by the same local control network and therefore can be added as the same communication node in WideQuick Designer. It can therefore be concluded that if multiple tags have a similar tag structure a variant of this solution would be useful in the WideQuick Water and Wastewater Treatment application.

Support objects are what allow a system integrator to display information about an object directly in the workview of layer 2 in the hierarchy. It has been requested that objects should also be able to show averages of a signal so to convey to the operator a normal range of operation for a certain time period. This function can be developed in different ways. One way would be to create an accumulative average value of all data points. This calculation for many objects might require extensive processing power especially if the value is to be updated frequently.

Another approach would be to only sum up values from the same point of the day for a certain time horizon, for example an hour back in time the previous two weeks. This would not only display more useful information for comparing the process during similar modes of operations but also decrease the size of each calculation. This in turn would enable more frequent updates.

5.3 Layer 3. The Process Unit Details and Support

It was decided that the third and forth level of the high performance HMI should be combined into a single layer. This was achieved using a single popup with several different tabs to control and overview all aspects of an object. This is a decision that the authors are satisfied with. It greatly simplifies the handling of objects. The customers were also positive to this. There are, however, some things to be added in the future, which is discussed in chapter 6.

WideQuick does not natively support maintenance management systems like machine cards. Our first idea was to store all machine information locally on the machine running the SCADA. The problem with this solution, however, is that any remote systems will not be able to access this file without using Remote Procedure Calls (RPC), which requires a higher license level of WideQuick to access these functions. An idea on how this could be solved is by using databases to store information about objects. In turn this would enable information to be requested from any system connected to said database. However, the security aspect of this has yet to be explored.

Feedback was given concerning the functionality of the history popup. The customers stated that they would like the view to automatically present data for a tag collected from more than one database so that a tag is displayed with the highest resolution possible but at the same time use the broadest time horizon available. An example for this would be if a signal has been logged to a database with low resolution for a year but only recently been included into a logger with a higher sampling rate. This could be solved with a script that polls the database for the current time horizon of the history object to check if there is data points available for the whole interval. If not go on to the next database connection that logs the tag, eventually choosing the one that best satisfies the condition.

5.4 Conclusions

The conclusions that can be drawn from this project are:

1. A standard is only really a standard if it is widely used in the industry.

2. A communication overview can be very useful when trying to troubleshoot the system.

3. Development in conjunction with the intended users yields the best possible application.

4. A communication overview is not possible to develop for system not running Windows since the currently available solution requires .NET support.

5. Further simplifications to the process of connecting tags to objects will reduce the initial setup time.

6. Having both layer 3 and 4 from the high performance HMI guidelines can be considered excessive for some scenarios.

7. Support objects that can show more advanced averages over some time horizon would provide the user of the application with more useful information than the real-time value. Allowing the user to more intelligently run the process. Implementing this in a smart way would reduce the computing needed.

6 Future Work

Below follows possible ways to continue this work after the thesis is concluded. The chapter has been split into customer driven and developer driven feedback to distinguish between what the customer has requested and what the developers would like to work on.

6.1 Customer Response Driven Work

Using the feedback from the demonstrations allowed for an updated list of requirements to be created. This is a combination of feedback both from the customers and internal feedback personnel at Kentima.

- Add the ability to define Connection, System and Device on a workview level, which means that every object can inherit these properties. This in turn will result in that the system integrator can save even more time while creating objects.
- Update/create a new Analog Value Display that instead of using percentages of min/max value to display warning/alarm instead takes a set value.
- Add a view for an overview of communication that does not rely on .NET plugins to ensure greater compatibility.
- More advanced analytical tools. For instance the ability to compare flows at a certain hour for a week.

6.2 Developer Reflection Driven Work

As developers of the application are more aware of certain limitations in the application than the customers might experience by just testing the software. For this reason a list of things that could be improved on to improve the stability and robustness of the application was compiled. Suggestions are listed below:

- Sub systems are as of today limited to a depth of a single directory, such that the system cannot be divided further into for example different processes of views of a processes that belong together.
- Update the analog value display to automatically retrieve min/max levels from the tag definition, further reducing the setup a system integrator has to complete, which in turn will result in a lower time to market.
- Finalize the map view to give additional geographic information about the plants.
- Since wastewater treatment is closely linked to weather (more rain means more incoming water) a smart weather view could come in handy. Possibly integrate this view with the map to show the forecast at each plant.

- Finalize the machine card common popup. This will give a native alternative to other maintenance software. This will also simplify calibrations as all calibration dates can be saved per object.

Bibliography

[1] "Sanitation fact sheet," https://www.who.int/news-room/fact-sheets/detail/sanitation, accessed: 2022-04-28.

[2] "Kepserverex," https://www.kepware.com/en-us/products/kepserverex/, accessed: 2022-05-05.

[3] "Mkdocs," https://www.mkdocs.org/, accessed: 2022-04-25.

[4] N. M. G. Cicceri, R. Maisano and S. Distefano, "A novel architecture for the smart management of wastewater treatment plants," *2021 IEEE International Conference on Smart Computing (SMARTCOMP)*, pp. 392–394, 2021-08-23 - 2021-08-27, Irvine, CA, USA.

[5] "Rening av avloppsvatten," https://www.vivab.info/vatten-avlopp/avloppsvatten/rening-av-avloppsvatten, accessed: 2022-04-28.

[6] "Rening av dricksvatten," https://www.vivab.info/vatten-avlopp/dricksvatten/rening-av-dricksvatten, accessed: 2022-04-28.

[7] S. D. Anton, D. Fraunholz, C. Lipps, F. Pohl, M. Zimmermann, and H. D. Schotten, "Two decades of scada exploitation: A brief history," *2017 IEEE Conference on Application, Information and Network Security (AINS)*, 2017-11-13 - 2017-11-14, Miri, Malaysia.

[8] K. Ashton, "That "internet of things" thing." Available at https://www.rfidjournal.com/that-internet-of-things-thing (2020/04/22), 2009.

[9] "Pera enterprise integration web site," http://pera.net/, accessed: 2022-04-28.

[10] "Isa112," https://www.isa.org/getmedia/626ce8b5-4d89-4924-98e4-082bc1c1982c/ISA112_SCADA-Systems_SCADA-model-architecture_rev2020-05-28.pdf, accessed: 2022-05-10.

[11] "What is hmi?" https://inductiveautomation.com/resources/article/what-is-hmi, accessed: 2022-04-27.

[12] B. Hollifield, D. Oliver, I. Nimmo, and E. Habibi, *The High Performance HMI Handbook*, 1st ed. Plant Automation Services, Inc., 2013, Brookfield, CT, USA.

[13] "Material for mkdocs," https://squidfunk.github.io/mkdocs-material/, accessed: 2022-04-25.

[14] "Wq va documentation," https://johanvikstrand.github.io/WQVADOC/, accessed: 2022-04-27.

Appendix

Appendix A - Interview Manuscript

This document contains the summarized answers to questions asked to four actors in the water and wastewater industry. The interviews were conducted by Markus Kvist and Johan Vikstrand for the master thesis project "General SCADA Application for Water and Wastewater Treatment". As water and wastewater treatment plants are deemed as protected objects all participants and their organizations have been anonymized and denoted as Customer A through D.

Question 1. Which HMI/SCADA solution are you using today, what do you like about it? What can be improved?

Customer A: Currently using Cactus Eye and are generally satisfied with the product. When asked what could be improved the answer was better handling of on-call procedures and a smart interface for easy configuration of who to contact concerning alarms and with what measures ensued.
Customer B: Uses a variety of different SCADA solutions depending on their customers requests. WideQuick is one of the solutions used. Generally the customer likes developer tools that enables the user to create unique objects as no plant looks the same. Time to market is the one most important parameter so any improvements that will reduce the time needed to configure the process is greatly appreciated.
Customer C: Uses iFix from General Electric, has some experience using WideQuick as well. Likes the visualisation tools from iFix and how it integrates well with their historian database and that a user can connect to a website to remotely access the SCADA. Their main concern is with the costs that this solution presents to smaller facilities where subscription models drive up the price when in reality very little functionality would suffice.
Customer D: Currently develops using WideQuick. Since there is no water and wastewater solution for WideQuick just yet they have resorted to using parts from WideQuick BMS. However WideQuick BMS lacks water and wastewater specific objects as well as analytical tools.

Question 2. Which objects are going to be needed?

Customer A-D: Requested the "normal" objects used in the water and wastewater industry. Would greatly appreciate the ability to design custom objects.

Question 3. How does the process look? How does the system architecture look?

Customer A: Operates several larger plants as well as many smaller ones including pumping stations. A SCADA is in operation today overlooking many of these processes.
Customer B: As they are a consulting firm they do not actively manage any plants. System architecture also depends on the customer currently employing consultants.
Customer C: Operates both water and wastewater treatment plants. Does however only want

to migrate the water treatment plants to WideQuick. Another system is already in place for wastewater.
Customer D: Operates both water and wastewater treatment plants. Want to migrate their entire operation to WideQuick.

Question 4. Are there any standards in use today? - Concerning the subject, tag structure, color coding and so on

Customer A: Has a company wide policy concerning colors. This has been subject to change several times and has therefore requested a simple interface to change the colors as needed. No other standards in use.
Customer B: No standard available. Every customer has their own standard.
Customer C: Not currently using any standard but wishes to change this in the future to achieve a more unified system.
Customer D: There was a standard in use previously. But as new employees have started and old ones have left this has been subjected to change and is no longer followed.

Appendix B - The full interview script

Disclaimer: The interviews were conducted in Swedish. This manuscript has been translated to English.

Which HMI/SCADA solution are you using today

- What do you like about it?
- What do you not like about it?
- What do you feel is missing in your current solution?
- What would make you switch and start using Kentima's Water and Waste Water solution?

Walk through of your process

- How does your system architecture look?
- Which processes are you monitoring?
- Which processes would you like to monitor in the future?
- What analysis are you performing on the data you collect from different processes and for what purpose?

I/O-list

- What object are needed in a water and wastewater application?
- What do you like about the objects you are using today?
- What do you not like about the objects you are using today?

Standards

- Is there any standard when it comes to tag structure?
- Is the any standard when it comes to color coding?
- Are there any rules or regulations that we need to keep in mind while developing the SCADA?