

Environmental reports from SQL

DAVID OLSSON

BACHELOR'S THESIS

DEPARTMENT OF ELECTRICAL AND INFORMATION TECHNOLOGY |

FACULTY OF ENGINEERING | LTH | LUND UNIVERSITY





Bachelor's Thesis

Environmental reports from SQL

By

David Olsson

Department of Electrical and Information Technology
Faculty of Engineering, LTH, Lund University

© Copyright David Olsson LTH

Davidolsson45@hotmail.com

+46727473300

Ingenjörshögskolan vid Campus Helsingborg

Lunds universitet

Box 882

251 08 Helsingborg

LTH School of Engineering

Lund University

Box 882

251 08 Helsingborg

Sweden

Printed in Sweden

Lund University

Lund 2019

Abstract

The main objective of this bachelor's thesis was to develop a prototype for a system in the shape of a database with applications for environmental reports from waste incineration plants. The intent is to create a showcase and a proof of technology for ÅF Malmö's customers who have an estimated benefit of this technology. The system needs to process data from a waste incineration plant according to relevant national and international regulations. Afterwards the data needs to be visualized with relevant parameters. The database is required to have support for real-time values that are used for alarms and plant overview.

Large parts of this thesis treats complex rules for various pollutions that require complex algorithms to distinguish possible excess emissions. Another large part has been the testing of useful methods and operations that are effective to manipulate data in an SQL-database and also presenting information from the database.

The results are a database-structure for emissions from a waste incineration plant and three applications for calculation, alerts, and generating reports. As the purpose of the prototype is to be able to implement on different distributed control systems the development of the database and application has considered adaptability for different distributed control systems.

Keywords: Environmental reports, PLC, SQL, Information handling, Database.

Sammanfattning

Syftet med examensarbetet var att utveckla en prototyp till ett system i form av en databas med applikationer för miljörapporter som inhämtar data ifrån styrsystem för avfallsförbränningsanläggningar. Avsikten är att prototypen ska visas för ÅF Malmös kunder som använder miljörapporter. Systemet ska bearbeta data enligt ett regelverk och sedan visualisera relevanta parametrar. Larm ska presenteras ifrån systemet då anläggningen överskrider ett gränsvärde för utsläpp.

Stora delar av examensarbetet har varit studier kring miljörapportering och testning av vilka operationer som är effektiva och användbara i mellan SQL-databasen och applikationerna. Resultatet av examensarbetet är en databasstruktur och 3 stycken applikationer för beräkning, larm och miljörapportering. Då syftet är att prototypen ska vara möjlig att implementera på olika styrsystem har applikationerna och databasen utvecklats med hänsyn till anpassningsbarhet för olika Distributed Control System.

Nyckelord: Miljörapporter, SQL, PLC, Informationshantering, Databas. S

Acknowledgement

I would like to thank Jan-Åke Roslund and Magnus Grenthe at ÅF for the opportunity, resources and guidance given to me during the execution of this thesis.

I would also like to thank Mats Lilja and Christian Nyberg from Lunds Tekniska Högskola for all the help and guidance during this thesis and for also giving me the tools and knowledge to be able complete this assignment. Thank you all!

Nomenclature

DCS – Distributed control system

SQL – Structured Query Language

PLC – Programmable logic controller

AMS – Automated measurement system

MSSQL – Microsoft SQL server

Contents

1. Introduction.....	10
1.1. Background.....	10
1.2. Aim	11
1.3. The Objectives of this thesis.....	12
1.4. Problem statements	12
1.5. Limitations	13
2. Technical background.....	14
2.1. Environmental Reports.....	14
2.2. Automated Measurement System (AMS)	15
2.3. Microsoft Visual C#.....	15
2.4. NO-SQL Database	16
2.5. EPPlus	16
2.6. Distributed Control System (DCS)	16
3. Methodology	18
3.1. Examination.....	18
3.2. Specification of requirements	18
3.3. Prototype architecture	19
3.4. Entity-Relation diagram.....	19
3.5. Testing	21
3.6. Evaluation.....	22
3.7. Source Criticism.....	23
4. Analysis and realization.....	24

4.1. Database Construction	25
4.2. Storing Data	25
4.2.1. Sensor Status	26
4.2.2. Normalization	26
4.2.3. Sensor Adjustment	27
4.3. Update process.....	28
4.3.1 Report update process	29
4.3.2 Feedback update process.....	29
4.3.3 Presentation process.....	29
4.4	29
4.4.1. Ten minute/half-hour reports	32
4.4.2. Day reports	35
4.4.3. Month/Year reports	36
4.5. Presentation	36
4.5.1. Summary half hour	36
4.5.2. Month Report.....	37
4.5.3. Environmental report.....	37
4.5.4. Year report	37
4.6. Operator Feedback.....	38
5. Results	39
5.1. SQL database	39
5.2. Applications	40
5.2.1. Update process.....	40
5.2.2. Presentation process.....	40
5.2.3. Feedback update process.....	41
6. Conclusions.....	43

6.1. Ethical dilemmas	45
7. Future Work	46
References.....	47
Appendix A: Example Reports	48

1. Introduction

This chapter gives a brief description of the background and purpose of this thesis.

1.1. Background

ÅF Industry Automation Division in Malmö delivers services to the industrial sector, infrastructure, and new constructions. The services vary from troubleshooting to overall solutions.

Multiple associates at ÅF have noticed a demand for reading and handling data from control systems at the customers plants.

A project was started at ÅF that enabled the possibility to create a simulated environment for a waste incineration plant and this created preconditions for a thesis examining storage and presentation from a waste incineration plant. A basic overview is presented in figure 1.

All waste incineration plants in European Union(EU) member states are required to abide by the European parliament and council directive for waste incineration [1]. The directive specifies minimum requirements for member states regarding waste incineration and under which circumstances incineration should be conducted.

Legislation is determined nationally and may result in harsher criterias, but nations can also apply for exceptions for individual plants. The Swedish parliament (Riksdagen) enacts the laws for waste incineration and Naturvårdsverket is responsible for enforcing the laws [1]. In the directive from Naturvårdsverket it states that automatic surveillance systems for measuring pollution levels is to be used [2]. The data must be stored and presented in an environmental report submitted to the authority annually.

Modern Distributed Control System (DCS) normally store historic data in local NO-SQL databases on-site and these databases are optimized for large sets of distributed data. When the data needs

significant manipulation, the NO-SQL database is insufficient. Therefore, a study on using SQL for data handling is needed.

1.2. Aim

The purpose of this thesis is to study which methods that are usable when creating/gathering data from a distributed control system regarding which is fastest and/or easiest. Another ambition is that the created prototype must be easy to implement on different DCS without the software being rewritten from the core.

The information presented from the prototype must be easy to understand and must not require any programming knowledge. The purpose of data presentation is to simplify and improve the efficiency of environmental reporting work and to create a material for analysis.

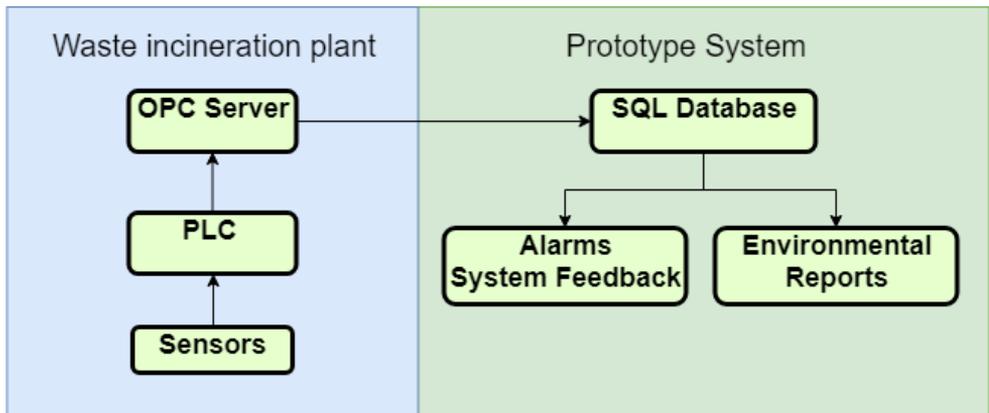


Fig. 1. A basic overview of system functions.

1.3. The goals of this thesis

The goals of this thesis are described in this subchapter. The goals were created in cooperation with advisors from both ÅF and Lund's University.

- Create a relational database for storing data from the waste incineration plants control system. Data stored must be minimal and relevant according to the desired prototype system and any changes to the database from users must be logged.
- Create applications for generating automatic environmental reports from the relational database. The applications must also present information for analysis of the plant historic performance.

1.4. Problem statements

During the construction of the prototype the problems listed had to be taken into consideration to avoid unnecessary restructuring of the database and applications. This thesis aims to present a solution for the problems.

- What numeric precision is needed in the database to calculate data without losing any information?
- What arithmetic value variation is presented from different sensors?
- Are different datatypes needed to fulfill the precision requirements?
- Which commercial applications are suitable for presenting the environmental reports?

- How can the adjustment and normalization of the digital measurements from the control system be converted to physical representative values?
- How should the database handle values that are incorrect due to calibration of sensors?
- Which parameters should be presented to the plant-operators in “real-time”?
- What should trigger an alarm and how does the application make sure the input values are not frozen?

1.5. Limitations

The focus for this thesis has been to create the database and applications required to present reports. Since a simulated test environment was not established due to company priorities, the raw data used in the database are produced by an application to mimic the values generated by the control system. The alarm function and real-time visualization have been prepared but not implemented.

2. Technical background

This chapter describes preconditions for this thesis and the software used. The rationale for decisions regarding methods and software choice is explained.

2.1. Environmental Reports

The purpose of environmental reports is to provide a basis for the supervising authority to oversee the regulated activity. The reports also give the plant owner grounds for how legislation demands are met. Self-monitoring can also be improved by using the reports. Portions of the reports regarding waste incineration are normally reported to an international organ. Example: the total CO₂ emission from Sweden's combined waste incineration plants is disclosed to European parliament [3].

The supervising authority for waste incineration in Sweden is Naturvårdsverket and they are responsible for enforcing that rules and regulations are complied with and met. An environmental report has many aspects and this thesis focuses on automated supervision and airborne pollutants.

In the beginning of each year afflicted waste incineration plants are required to submit an environmental report and disclose any violations to Naturvårdsverket. A violation occurs when a measured pollution exceeds the prescribed limit value for the significant pollution. The violation limit can be divided into:

- Day average
- Half-hour average
- Ten-minute average.

A day is divided into 48 half-hours and 144 ten-minute intervals, an interval can have a percental limit and an absolute limit. A percental limit may be exceeded but repeated violations can result in fines.

2.2. Automated Measurement System (AMS)

Waste incineration plants with capacity above 50 MW typically use an automated measurement system to gather and store data from sensors in the incineration process. Modern DCS have the possibility to store large amounts of data and the AMS select which sensors are to be monitored (see figure 2).

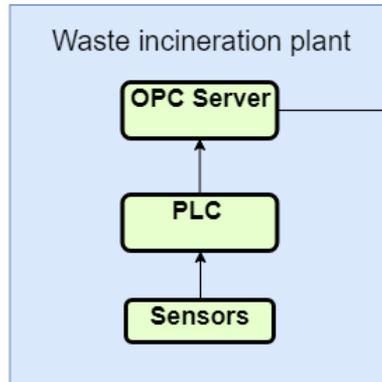


Fig. 2. Basic overview of the automated measurement system.

The measurement process are inspected in three stages Quality Assurance Levels(QAL) 1-3 according to the standard SS-EN 14181:2014 [8]. QAL1 is an insurance that the AMS is installed properly and the plant supervisor is responsible for conducting the inspection. QAL2 is an inspection and a calibration of the plant sensors related to the AMS. The QAL2 is typically conducted every five years but may be more frequent on behalf of the supervising authority. QAL3 is an inspection to make sure the sensors are within approved error margin range. QAL3 has to be conducted at least once a quarter.

2.3. Microsoft Visual C#

C# is multipurpose programming language developed by Microsoft in the early 2000s. C# requires the .Net framework for compiling program code. Functions for communicating with both SQL databases and MS Excel is incorporated in the C# standard library.

The reason C# was selected for application development is that the ABB DCS 800xA has support for executing C# applications on the server end. This leads to a smaller and more agile prototype system.

2.4. NO-SQL Database

The key difference in NO-SQL Databases compared to SQL is that the data stored is not relational. This has advantages to scaling the databases to handle larger sets of data. Read/write operations are faster than a relational database, however complex operations are more difficult. [4]

2.5. EPlus

Eplus is a library used in C#, used to create Excel spreadsheets from program code. Eplus write directly into the spreadsheet in contrast to the standard library that starts a background process for writing to a document. The results is a fast write-time. [5]

2.6. Distributed Control System (DCS)

A distributed control system contains more than one computer, controller, or PLC. A human-machine interface (HMI) is placed centrally and can supervise/control the system from another geographical place. In modern DCS, bulkdata from I/O, signal values, and statuses are stored in databases.[6]

2.7. Microsoft SQL server (MSSQL)

MSSQL was chosen because the ability easily interact with other Microsoft software. For example Power BI has the ability to present data from a SQL server and since both are Microsoft products the interaction was believed to be more efficient compared to third party software.

2.8. Sensors

The sensors sending information regarding emissions to the automated measurement system need to be certified by TUV, MCERTS or other certification body [7]. The purpose being quality assurance of the plant measurements. The signals received by the prototype are simulated and the simulated values are randomly generated between different ranges based on a datasheet from a working waste incineration plant. The following sensors are simulated:

- CO – Carbonmonoxide
- HCl – Hydrogen Chloride
- NO_x - Nitrogen Oxide (NO/NO₂)
- SO_x – Sulfuric Oxide(SO₂/SO₃)
- STOFT – Dust
- TOC – Organic Charcoal
- TEMP – Temperature
- H₂O – Humidity
- PRESSURE – pressure
- O₂ – Oxygen level

3. Methodology

This chapter describes in chronological order what methods were used to create and test this thesis prototype.

3.1. Examination

Initially the priority was to create clear and absolute definitions and rules for the environmental reports. The national legislation regarding waste incineration was studied with reference to the EU Directive. Afterwards studies were made on current environmental reports in order to see the practical implementation of the legislation. User requests for system functions were also considered. The user request was provided by ÅF on behalf of the customers. An example of a request is the trends that show the current sensor-value compared to the different limit-values.

3.2. Specification of requirements

The information gathered during the examination phase was summarized into a list containing the specification of requirements as described in this section.

- Sensor signals from the control system are to be stored unaltered in the database with the parameters: name, value, and status.
- Constants, limits, and authors are to be stored in the database and the point in time for changes made.
- 10 minutes-,30 minutes-, and day-mean values are to be stored in the database.
- User made modifications to reports must be logged.

- Reports must have access to all normalized and compensated mean values.
- Report can be presented between user specified dates.
- Alarm trigger when pollution limit is exceeded.
- Alarm trigger when sensor status is set to nonfunctional.

3.3. Prototype architecture

When the prototype requirements had been established, the next step was to find a model for processing incoming data for storing and presentation. The aim was to use the strengths of the different software in order to fulfill the requirements. Software from different developers were compared regarding price, speed, compatibility, and convenience. During the project the architecture had to be updated, in order to solve unforeseen problems and conflicts.

3.4. Entity-Relation diagram

An entity-relation(E-R) diagram was created to implement a base for testing phase. The E-R diagram was the first step before constructing the database, in order to create a basic structure of the data. The initial E-R diagram was designed in order to handle most of the calculations in the database, but this was changed later in the project and is explained in 4.3.

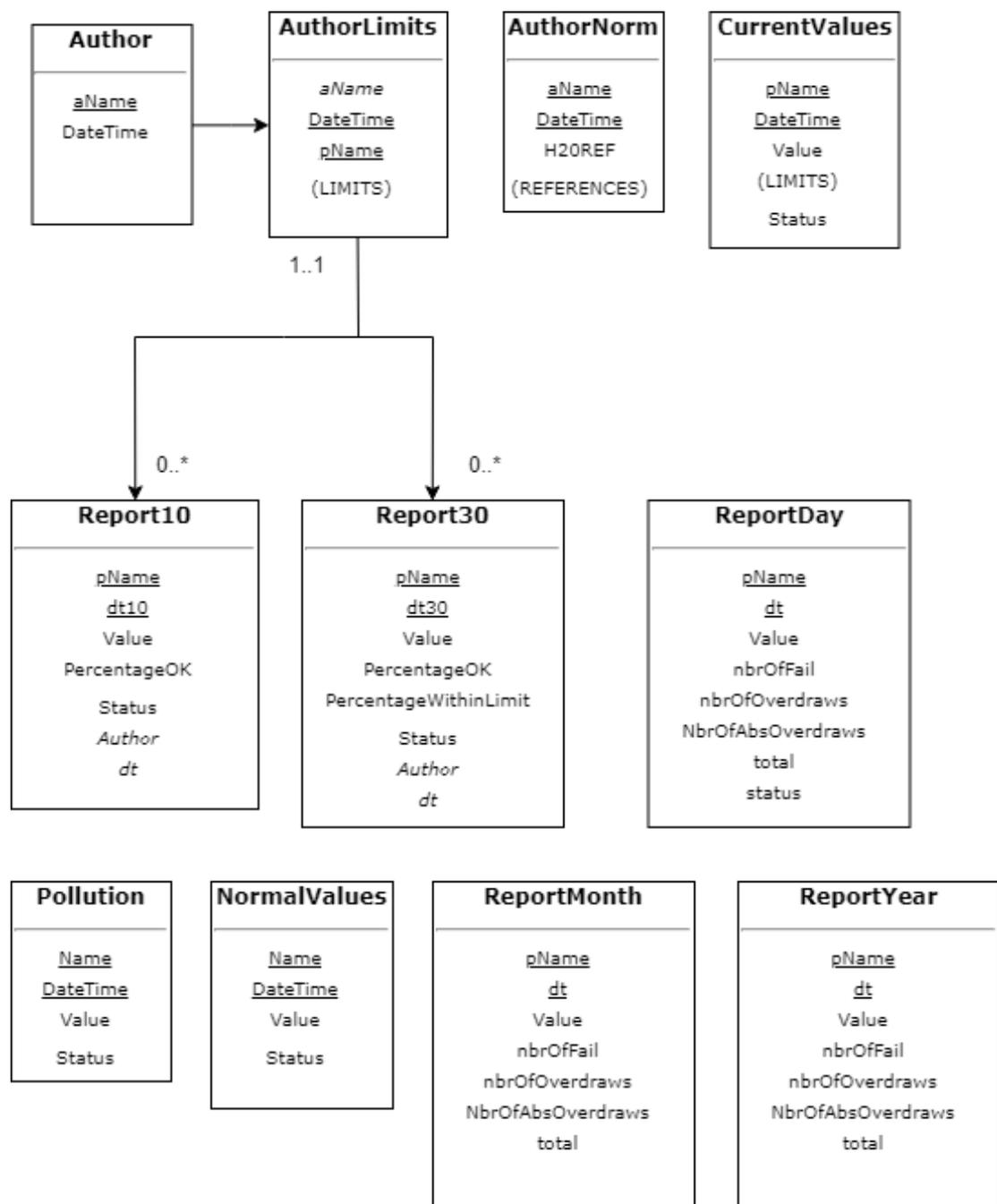


Fig. 3. E-R Diagram

Figure 3 is a visualization of the final database structure. Parentheses are used to summaries the similar attributes in the entities.

3.5. Testing

The testing phase was divided into multiple stages, the purpose was to simplify the development of the applications and the database structure.

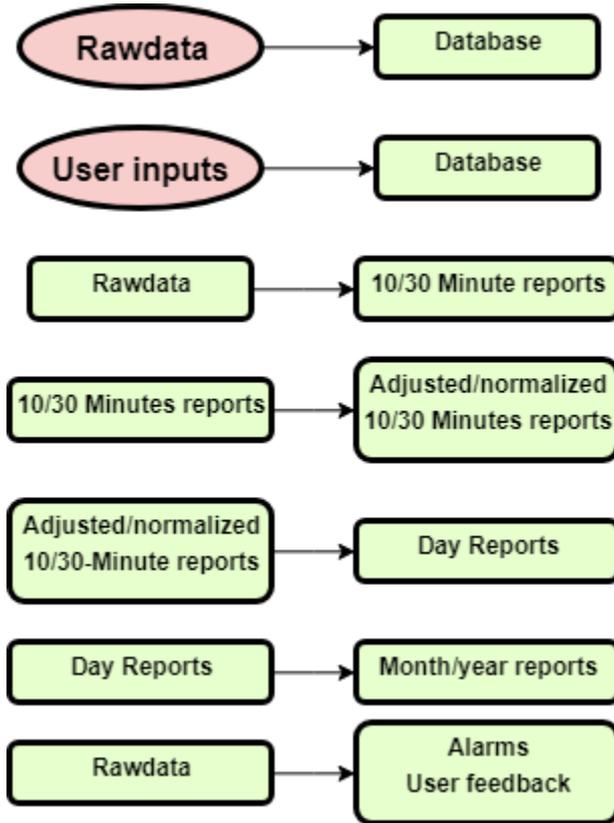


Fig. 4. Overview of simplified prototype development stages.

Testing involved construction of applications used to check speed and functionality. Other features such as disk usage was also considered during this process. The sequences in figure 4 are a description of

which order the prototype was developed. Deviations occur when changes to earlier development stages needed to be made. An example being the user inputs needed to process a day report differ from the 30 minute report and changes to the prototype have to be implemented.

3.6. Evaluation

After eight weeks, during the testing phase a presentation was prepared and held for the team related to waste incineration plants at ÅF. The purpose of the presentation was to inform what had been discovered during the examination phase and to show how far the thesis had progressed. After the presentation, the different advantages regarding solutions for data manipulation were discussed. Which prototype functions were most important to complete were also discussed. The conclusion was to focus on data presentation from the prototype.

During the last week at ÅF a final presentation was held to present the finished prototype for the team. Afterwards improvements and future work were discussed.

3.7. Source Criticism

The information regarding legislation and methods for measuring pollutions that are references to a government website was considered to be trustworthy and most of the information has been cross referenced between different organs. For example, the pollutants that are subject to continuous surveillance is listed in EU directive, decree (2013:253) by Regeringskansliet, and in Naturvårdverkets regulations. But since changes are possible to these rules, it is possible that that some parts described in this thesis may be outdated later on. This regards the following sources: Regeringskansliet [1], Sweden, Naturvårdsverket[2], Sweden and European Parliament[3]. When cross referencing older documents regarding waste incineration the changes were minor. Other references that describe functions of software have been crossreferenced and also the functions have been tested to work in the way described to some extent.

4. Analysis and realization

This chapter presents the stages of development for the prototype system.

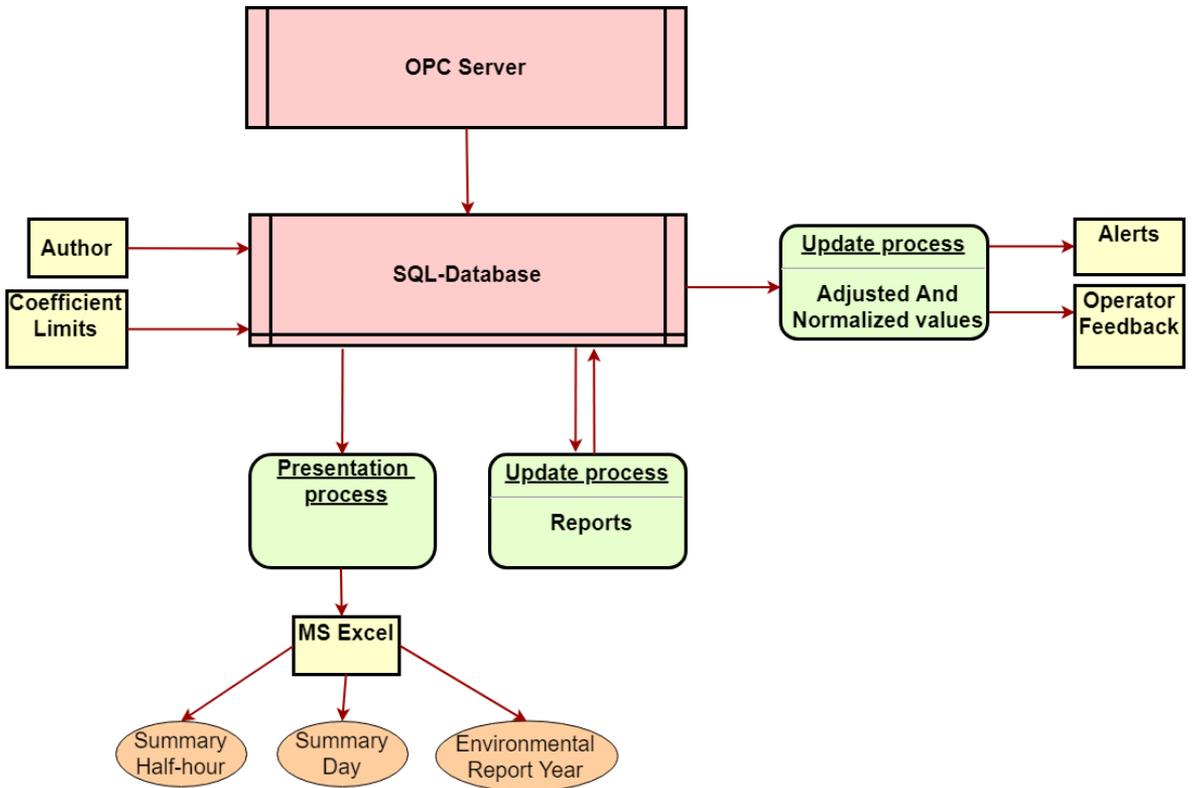


Fig. 5. Overview prototype dataflow.

Figure 5 is a description of the different processes managed by the prototype and in which direction data is transferred.

4.1. Database Construction

Microsoft SQL database server was installed on the project computer and initially the theoretical E-R diagram (see figure 3) was implemented. The ambition was to run calculations and modifications using SQL-queries. The advantages being that human errors in system architecture would be minimized when using a well-tested and already implemented software. The system process speed was predicted to be high since the database uses advanced algorithms for storing and accessing data.

The Microsoft SQL Server has a built-in function called Server Agent that can perform complex operations and queries and has several options for executing commands at a specified point in time. This function is ideal since portions of the calculations only needs to execute after the time has passed certain intervals.

Disadvantages of database calculations is the possibility to sequentially follow the steps made by the software, compared to a high level programming language.

4.2. Storing Data

The data from the control system is stored unaltered in the pollution table. The table contains:

- Name – the name of the sensor, preferably simplified for comprehension
- Datetime – the time and date for the data insertion.
- Value – the unaltered sensor value
- Status – status of the sensor value

The pollution table contains the base data for all prototype operations. Entries into this table are entered once per minute, this frequency was selected since all requirements for the environmental reports are met and the disk usage is low compared to a higher frequency. Other advantages are faster execution times and easier troubleshooting.

Disadvantages include lesser calculation precision, due to the mean value is based on fewer values. The operator feedback can only be updated at the same frequency as the data is entered into this table (see 4.6).

The amount of inputs per minute into the pollution table is limited by the table size restrictions by MS SQL and the network transmission speed. Changing the input frequency has no impact on the prototype performance (see 4.7).

4.2.1. Sensor Status

Sensor status is a representation of the waste incineration process surrounding information. The status of the sensor is categorized into three states: **Valid**, **Sensor Malfunction**, and **Valid Defect**.

Valid Defect status is entered when the incineration startup process begins and therefore pollutant measurement is disregarded in accordance to regulations. The status can be set by operator during sensor tuning.

Sensor Malfunction status is set when unreadable signals are received or other system functions indicate sensor malfunction.

Valid is set when all requirements are fulfilled and the sensor values are measured under the proper conditions.

4.2.2. Normalization

The normalization process for sensor values refers to transforming a value received during certain conditions to a common reference.

Pollutants concentration is normalized in accordance to temperature, pressure, oxygen content, and H₂O content. Depending on what type of fuel is incinerated, fluctuations occur in the measured sensors. In order to compare the released pollution to a limit, the value and the limit must be compared during the same conditions. A normalization

coefficient is formed by measuring the current value of the variables and comparing them to the referenced value. The equation used is:

$$K_{\text{Normal}} = K_{\text{H}_2\text{O}} \times K_{\text{O}_2} \times K_{\text{Temp}} \times K_{\text{Pressure}}$$

$$K_{\text{H}_2\text{O}} = \frac{100 - \text{H}_2\text{ORef}(\text{Desired humidity})}{100 - \text{H}_2\text{OSensor}(\text{Measure Humidity})}$$

$$K_{\text{O}_2} = \frac{21 - \text{O}_2\text{Ref}(\text{Desired oxygen concentration})}{21 - \text{O}_2\text{Sensor}(\text{Measured oxygen concentration})}$$

$$K_{\text{Temp}} = \frac{273 + \text{TempRef}(\text{Desired temperature})}{273 + \text{TempSensor}(\text{Measured temperature})}$$

$$K_{\text{Pressure}} = \frac{\text{PressureRef}(\text{Desired pressure})}{\text{PressureSensor}(\text{Measured } \textit{pressure})}$$

By multiplying the measured sensor value with the calculated normalization coefficient, the new value is considered normalized.

Standard reference values are:

- Humidity reference: 0%
- Oxygen reference: 11%
- Temperature reference: 273 degrees Kelvin
- Pressure reference: 101.3 kPa

4.2.3. Sensor Adjustment

Adjusting the stored sensor values in the prototype database is required in order to be able to tune sensors for greater accuracy. These adjustments must be traceable in order to revert any changes done to the sensor value. Storing the added constants provides a

possibility for transparency in the calculations. For example when new sensor is installed in the plant, the sensor can be adjusted in the prototype system and not in the control system. This also gives the benefit of being able to make changes to earlier sensor values.

Adjustment calculation:

$$\text{Value}_{\text{Adjusted}} = \text{Value}_{\text{Raw}} \times \text{Coefficient} + \text{Offset}$$

4.3. Update process

During the initial testing phase, the process of creating half-hour summaries using only SQL-Commands proved difficult due to the fact that temporary variables are needed for calculations. The different statuses and values have to be compared in order to determine if the average value is viable. Although temporary variables are supported in MS SQL, the implementation was deemed too time consuming to implement. Therefore, a new approach was decided in which an application generated the data in the database. The application was implemented using C# and makes the bulk of calculations and comparisons. Only two datatypes were used, namely 'Int' and 'Decimal'. 'Int' is used for all variables with integer variables and 'Decimal' datatype is used for decimal values.

The 'Decimal' datatype uses up to 17 bytes and this gives additional space for upgrades. Based on signals found on two separate waste incineration plants, the value was between 0-1500 for all sensors. Sensor precision vary between 0 to 4 decimals. This means that the datatypes selected is predicted to be able to handle all system inputs.

Numeric datatypes in SQL databases are either defined as 'approximate' or 'exact' and by using exact datatypes the accuracy can be changed by the system architect in order to avoid information loss. The downside to using exact datatypes is larger disk usage.

4.3.1 Report update process

Most data handling is conducted in the report update process application. The data stored in the database is read and transformed to reports. Due to many read/write operations conducted between the process and the database, the process is recommended to be executed every thirty minutes in order to keep minimum influence on installed system. It is possible to perform a report update retroactively on a system with the current prototype version, but it may result in a significant computing time.

The report update process creates and stores: Ten minute/half-hour reports, day reports, month reports, and year reports.

4.3.2 Feedback update process

The frequency of which sample are gathered from the control system and stored in the prototype database is the highest which the database can update parameters presented to an operator. The data stored from the control system is unaltered and can be presented in such a raw format. Therefore, the information needs to be processed before presented. This is handled by the feedback update process. The process occurs once a minute by default.

4.3.3 Presentation process

With the absence of a graphical user interface the reports have to be created automatically. Therefore, the presentation process was implemented, when executed it creates MS Excel-documents for the months/years for which data is accessible. An interface for presenting a summary for days/weeks is available but not included in the prototype.

4.4 Data Manipulation

Described briefly in 2.1 the requirements for the environmental reports are elaborate. Different pollutants may differ both the limit value and in which intervals that are to be measured. However, the core pollutants have identical measurement intervals and this provided the base for the initial prototype. For example all pollutants need to be evaluated every 30 minutes, but carbon monoxide also needs to be evaluated every 10 minutes.

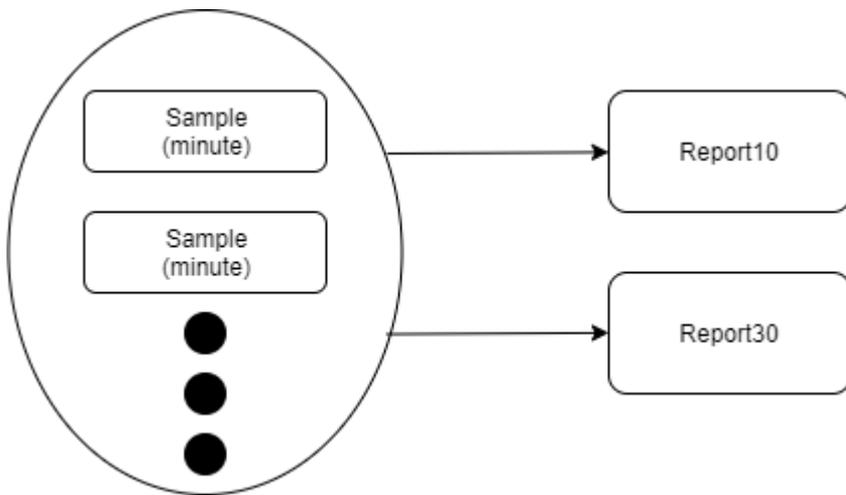


Fig. 6. Signal values to base reports.

The base reports contain an average value formed from the samples from the past 10/30- minutes. The samples are an average of the sensor values for 60 seconds.

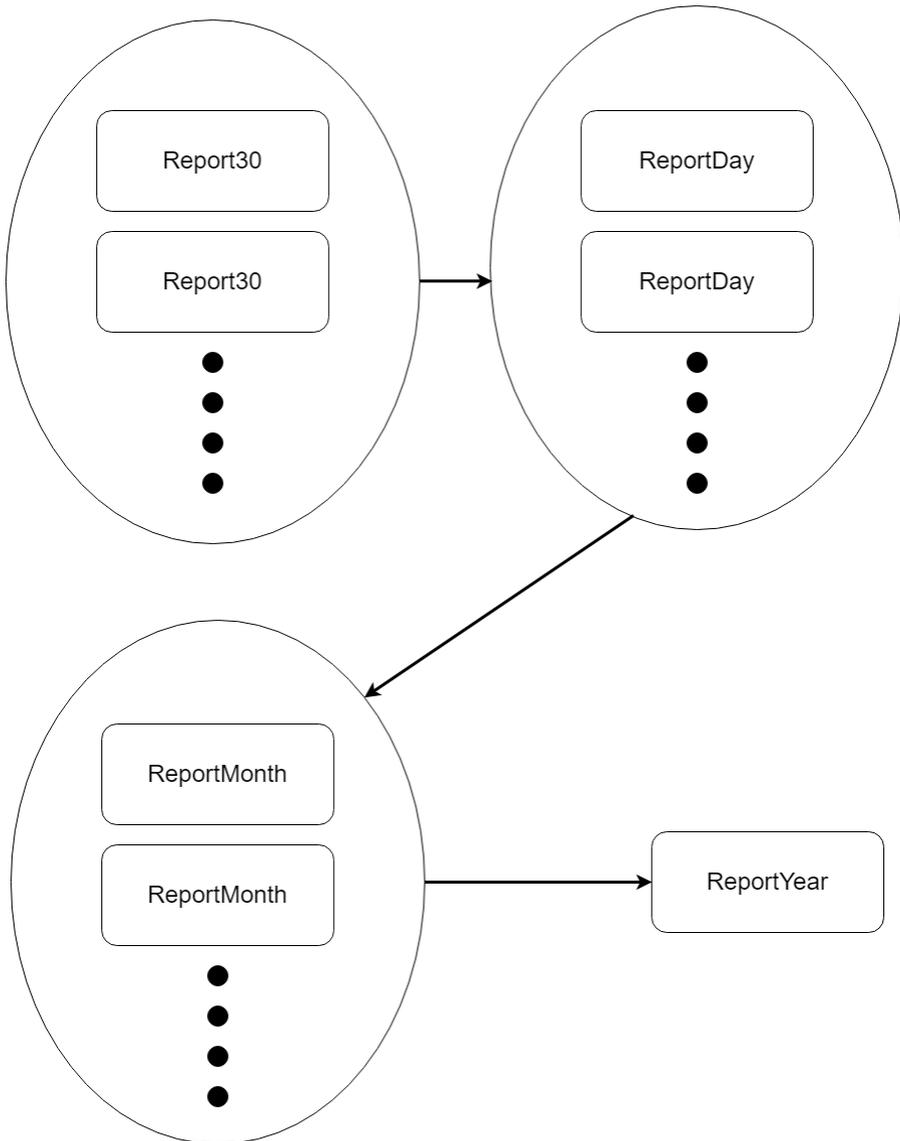


Fig. 7. Relation overview for reports.

A report contains a summary for the implicated interval.

4.4.1. Ten minute/half-hour reports

Report30 table in the prototype database contain:

- pName – Pollutant name, a reference to the which sensor the report represents.
- dt30 – time and date representing the past 30 minutes.
- Value – Average value for the interval.
- PercentageOk – A percentage describing the number of values which are valid versus the total number of values.
- Status – the label representing the interval summary.
- Author/dt – a reference to the which authorlimit applies during this interval.

The half hour reports and ten-minute reports are handled and stored identically the only difference being the time interval. This section refers both reports but uses the half hour report as an example.

Half hour reports represent a summary of the sensor measurement made during a thirty-minute period. The report contains the average value based on all signals that are valid and a status description for the summary (see figure 6).

When the update process triggers the “create half hour report”-function, the application retrieves an average value for sample-entries for the previous thirty-minute interval which only include the valid entries. The emission limits and adjustment coefficients added by the active author are retrieved and a normalization coefficient is calculated from the database.

When these variables are gathered the average value can be adjusted and normalized (see figure 8). The percentage of valid sample entries is calculated and if this is less than 60%, the report is considered to be nonfunctional and the average value is not relevant. If the valid sample entries are more than 60% then the average value is compared to the pollutant limits. Afterwards the status is calculated and the

adjustments made to the average value is reverted. This means that the average value is considered normalized and not adjusted.

The adjustment constants and pollutants are stored in the authorlimits table. This table regulates all changes made to the raw data and stored reports in the database. The table is linked to the reports with a key and when a report is read authorlimits is also read to get adjusted values. Another solution is to store the reports adjusted and update the report when a change has occurred.

This solution was tested and the result was that it is possible but time consuming. This is because the “create half hour report”-function must be executed once for every changed report.

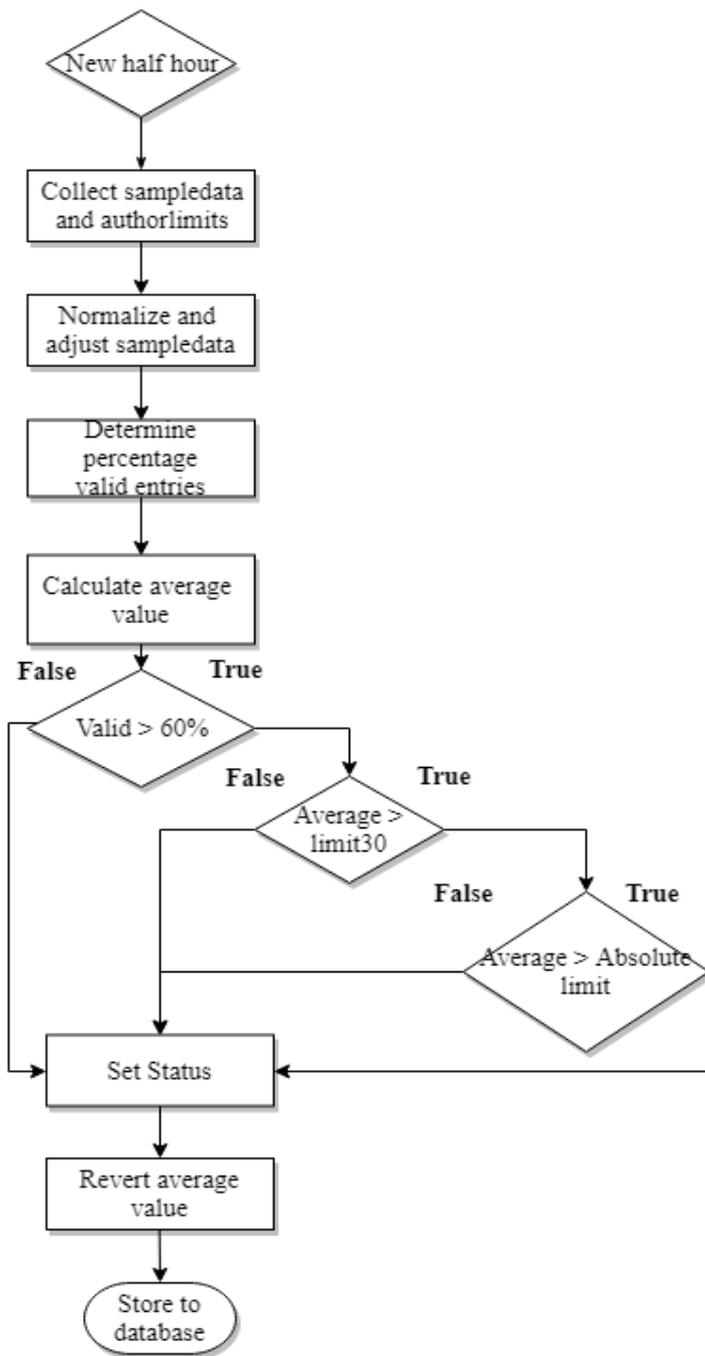


Fig. 8. Half-hour algorithm
 Overview when creating a half-hour report.

4.4.2. Day reports

ReportDay table in the prototype database contain:

- pName – Pollutant name, a reference to the which sensor the report represents.
- dt – Date representing the past day
- Value – Average value for the interval.
- nbrOfFails – The number of half hour intervals disregarded due to error rate.
- nbrOfOverdraws – The number of half hour averages that exceeded pollutant limit value.
- nbrOfAbsOverdraws – The number of half hour averages that exceeded pollutant absolute limit value.
- Total – total number of half hours excluding the status valid defect.
- Status – the label representing the interval summary.

Day reports store all relevant data gathered from the control system during the last 24 hours. The average value is normalized and adjusted to promote fast access to the report table.

When the update process triggers the “create day report”-function the application calculates an average value of the half hour reports for previous 24 hours including only the valid reports. The average value calculated from is based on all half hour that are adjusted with the individual limits based on the relation of the report and the authorlimit table.

The total number of half hour reports that are nonfunctional, limit value overdrawn, and absolute limit value overdrawn is counted. The status of the day report is determined by; in overriding order: number of nonfunctional reports > 5, average day value > limit value, previous statements false. When the function calculations are completed the report is stored in the database.

4.4.3. Month/Year reports

The month/year reports created by the report update process is a summary of day reports (see figure 7). The tables have identical attributes as the day reports except for the average value which has been excluded because this attribute is not required by regulations to present. The purpose of month/year reports is to simplify the process of presenting the data from the database without using elaborate queries, another advantage being increased speed reading from the prototype database. The disadvantage being more disk space is used although it is small compared to other database tables.

4.5. Presentation

In order to create direct and understandable reports presenting the information gathered by the prototype system, the ambition was to collaborate with users and operators of similar systems. But due to other project priorities the target company could not facilitate this request. Therefore the foundation for the presentation were founded by studying similar reports and cross-referencing the information with regulations. The design idea was to make the presentations to include all information that is required by regulations without making the presentation overwhelming. (See appendix B Example Reports)

Microsoft Excel was chosen as the template for the reports, this is because it is a commonly used application that many users are comfortable with. Excel has many useful functions and has regular maintenance and updates compared to competitors.

4.5.1. Summary half hour

The purpose of the half hour summary is to easily be able to backtrack when an overdraw or error has occurred. The summary (appendix A) presents all the information stored in the report30

between two user defined dates. The information is grouped and sorted by time and if any violation has occurred during a 30-minute period that row is highlighted with red color. The report also features a graph over the selected time period in order to analyze the sensor values leading up to the event and also after the event.

4.5.2. Month Report

The month report presents an overview of the values for the selected month. Each average day value for every pollutant is presented along with the related limit value. Any average day value overdraw is highlighted. A summary for all half hour reports and ten-minute reports is presented. The report is created and stored on a drive specified by the administrator installing the application on the workplace.

4.5.3. Environmental report

The final environmental report presents an overview of the values gathered from the control system during the last year. An effort is made to make the visualization similar to the month report in order to make the information searched for easy to find.

4.5.4. Year report

The year report is a complement to the environmental report, and the purpose is to summarize the information from the environmental report for an export to the supervising authorities application. The data is presented in order by pollutant then in order by rule paragraphs.

4.6. Operator Feedback

Generating the data to produce alerts and live feedback from the database, required an application that present the signal values at a rate that was adequate. The signal values must represent the actual physical quantity not the raw value. The highest frequency of this application is limited by the frequency of which the raw data is added to the database. Since the prototype adds raw data once per minute, the feedback application was also tested with this rate.

The application gathers data from the prototype database and stores the needed data in the 'CurrentValues' table. In the current version the normalized and adjusted value, pollutant limit value, and status are stored in the table. If a value gathered from the OPC server has not changed in the last hour, the status of the value is set to 'Frozen'. This function is requested by ÅF because of a repeating occurrence of frozen values from the NO-SQL database due to software issues.

Alerts can be generated simply by comparing the values stored in the 'CurrentValues' table.

5. Results

This chapter presents the results and the final components of the prototype.

5.1. SQL database

The database construction is the central part of the prototype because all information has to pass through it. All necessary data is stored in the database tables.

Disk Usage by Table [master]

on AF-L06976\TESTBASE at 4/26/2019

This report provides detailed data on the utilization of disk space by tables within the Database. The report does not provide data for memory optimized tables.

Table Name	# Records	Reserved (KB)	Data (KB)	Indexes (KB)	Unused (KB)
dbo.author	2	16	8	8	0
dbo.authorLimits	20	16	8	8	0
dbo.authorNorm	1	16	8	8	0
dbo.currentvalues	6	16	8	8	0
dbo.MSreplication_options	3	16	8	8	0
dbo.normalValues	4 204 800	228 424	226 832	1 520	72
dbo.Pollution	6 307 200	335 624	333 400	2 160	64
dbo.report10	63 559	6 664	6 576	48	40
dbo.report30	129 850	12 104	11 984	88	32
dbo.reportDay	3 163	328	312	16	0
dbo.reportMonth	98	16	8	8	0
dbo.reportYear	7	16	8	8	0

Fig 9. Disk usage for 2 years ordered by table.

Figure 9 shows the table containing the sample data contain 30 rows for each 30-minute summary and this pattern is applied for every increasing interval. The used disk space also correlate to the number of entries per table. Small variations occur depending on how much information is stored per row.

5.2. Applications

Three prototype applications were created during this project and each application handles different tasks. The applications are tested on the computer used to develop the applications. Using Windows task scheduler, applications are executed once every day.

5.2.1. Update process

Executed once every half hour the update process calculates the necessary functions needed to present reports. Benchmark test were made on the laboratory computer and the results were:

- Creating one 30-minute summary: 285 milliseconds
- Creating one day summary: 2 milliseconds

The significant difference between execution times are explained by different sizes of tables leading to a query conducted on the larger table require more processing time. Note that the benchmark test was conducted on the laboratory computer and this has impact on the result since the database normally is hosted on a separate server and network latency has an impact on the results. Also, the results depend on the hardware used by the computer/server running the application.

5.2.2. Presentation process

The presentation process is executed once every month and results in the creation of a monthly report. When a year has passed two reports summarizing the year are created. The first one is the environmental report which contain detailed overlook of plant emissions. The second one is the year report, which is a summary of violations that have occurred during the past year and that are required to be submitted to the supervising authority.

5.2.3. Feedback update process

Creating and storing sensor-values that are normalized and adjusted is a feature needed in order to monitor the incineration process. The current version of the prototype stores the factual sensor-value along with the limits related to the sensor. This way the value is easily compared to the limit. This process can be expanded to store and process data in order to show specific trends.

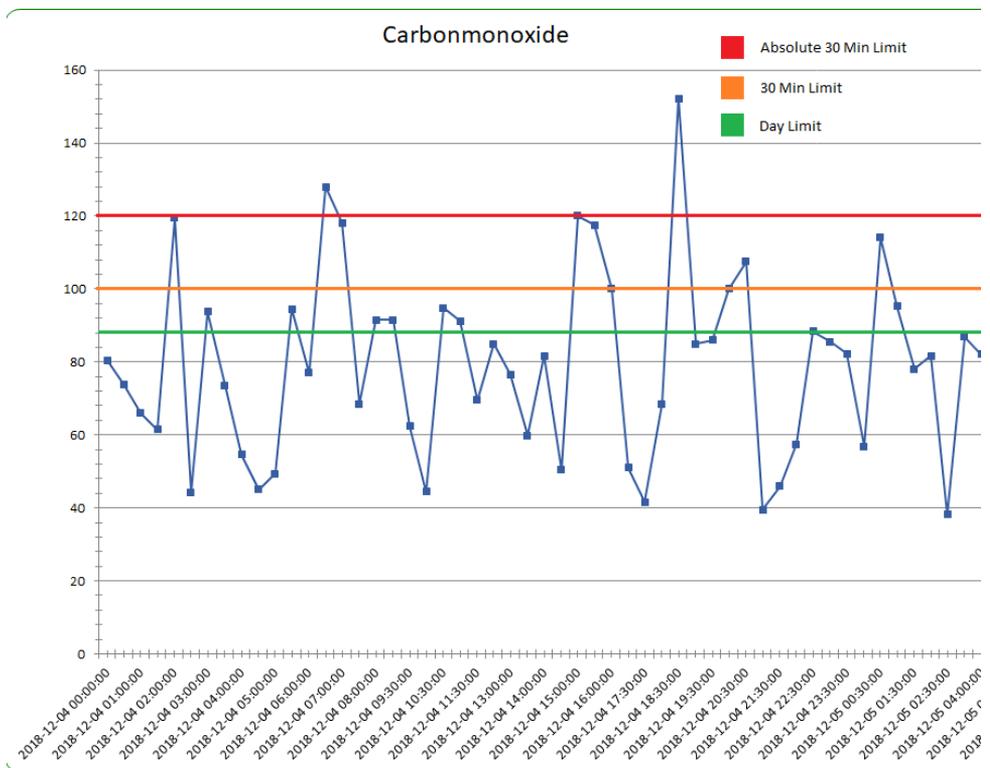


Fig 10. An example of trends retrieved from the prototype.

Figure 10 show the half hour mean values for carbon monoxide. A couple of violations has occurred and any violations for the 30 minute limit results in fines. Violations to the absolute limit also results in fines and will most likely require the plant to shut down. The day

mean value is not calculated but the trend is showing that the plant is operating close to the day limit.

5.3. Scalability

The thesis prototype has been developed to be able to simply add new pollutant sensor values and limit values. No other changes need to be implemented to provide results. During the testing phase, the system was upscaled with a factor approximately 300% and the performance was not significantly altered. The prototype should however be tested with more sensors before the scalability can be properly defined.

5.4. Reliability

This prototype was intended to be a proof of technology and before the prototype can be used an extensive test of the software must be conducted. Functions should be implemented for detecting errors in the database.

6. Conclusions

The objectives of this thesis were to create a relational database and the necessary applications needed to store and handle information from a waste incineration plant. The objectives were completed and as a result the prototype proves the possibility to create advanced calculations from the control system via a SQL Database. The main advantages using this type of database structure include: scalability, reliability, and cross platform support. The starting premise was that the system would be tested using real data collected from a waste incineration plant, but this could not be achieved due to other priorities at ÅF.

- What numerical precision is needed in the database to calculate data without losing any information?

Numeric datatypes in SQL databases are either defined as ‘approximate’ or ‘exact’ and by using exact datatypes the accuracy can be changed by the system architect in order to avoid information loss. The downside to using exact datatypes is larger disk usage. Only two datatypes were used, namely ‘Int’ and ‘Decimal’. ‘Int’ is used for all variables with integer variables and ‘Decimal’ datatype is used for decimal values. The ‘Decimal’ datatype uses up to 17 bytes and this gives additional space for upgrades.

- What arithmetic value variation is presented from different sensors?

Based on signals found on two separate waste incineration plants, the value was between 0-1500 for all sensors.

- Are different datatypes needed to fulfill the precision requirements?

Only two datatypes were used namely 'Int' and 'Decimal'. 'Int' are used for all integer variables 'Decimal' datatype is used for decimal values.

The 'Decimal' datatype uses up to 17 bytes and this gives additional space for upgrades.

- Which commercial applications are suitable for presenting the environmental reports?

Microsoft Excel was chosen as the template for the reports, this is because it is a commonly used application that many users are comfortable with. Excel has many useful functions and has regular maintenance and updates compared to competitors.

- How can the adjustment and normalization of the digital measurements from the control system be converted to physical representative values?

Mean values are stored normalized and when the values are read, the prototype adjusts the value that is presented.

- How should the database handle values that are incorrect due to calibration of sensors?

Values that are labeled 'Calibration' are stored but do not contribute to the mean value.

- Which parameters should be presented to the plant-operators in "real-time"?

Since an interview with a user regarding waste incineration was not conducted, the current presentation show the current pollutant compared to the pollutant limits.

- What should trigger an alarm and how does the application make sure the input values are not frozen?

6.1. Ethical dilemmas

The purpose of the function to retroactively be able to change stored values in the database, was to be able to change inaccurate values gather by a faulty sensor. This seems like a quite necessary function because if the error is undetected for a longer period it may lead to unusable data and possibly fines. But the function may also be subject to mistreatment to edit the plant statistics. The prototype has a function to track who has made these types of changes and during which time. But unless the system is properly supervised, mistreatment can occur. One solution to this problem may be keeping records on all sensor exchanges in the plant.

7. Future Work

The next logical step would be to connect the prototype to a real OPC server and configure and test the software. Afterwards displaying feedback information should be tested. The result gathered from the prototype should be presented for regular users of the similar systems for feedback and improvement. Since all data is stored in a SQL database, opportunities for presenting data in different ways and on different platforms is increased. For example, the feedback given to the plant operator may be altered to present relevant information to the plant supervisor. Also using cloud-based solutions to present data on mobile devices may be a desirable function.

New measuring procedures for other pollutants are being developed to meet with EU demands and once these procedures are tested and validated, surveillance systems must monitor these pollutants. Preparing for these changes could be a good way to upgrade this prototype.

References

- [1] **Regeringskansliet, Sweden.** *Förordning (2013:253) om förbränning av avfall.* <http://rkrattsbaser.gov.se/sfst?bet=2013:253>
- [2] **Naturvårdsverket, Sweden.** *Naturvårdsverkets föreskrifter om miljörapport*
<https://www.naturvardsverket.se/Documents/foreskrifter/nfs2016/nfs-2016-8.pdf> [Accessed: 2019-06-23].
- [3] **European Parliament, (EG) nr 166/2006**
<https://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2006:033:0001:0017:SV:PDF>
[Accessed: 2019-06-23].
- [4] N.Leavitt. *Will NoSql databases live up to their promise?*
IEEE Computer Society, 0018-9162, February 2010
- [5] **J.Källman.** *Epplus Readme*
<https://github.com/JanKallman/EPPlus/blob/master/README.md>
[Accessed: 2019-06-23].
- [6] **B.R. Mehta, Y.J.Reddy.** *Industrial Process Automation Systems*
Butterworth-Heinemann, 978-0-12-800939-8, 015
- [7] **LAND Instruments International.** *EN1481 – Information*
http://www.lirkorea.com/Landinstruments.net%20Website/combustion/downloads/pdf/EN14181_information.pdf
- [8] **Svenska Institutet för standarder**
<https://www.sis.se/produkter/miljo-och-halsoskyddsakerhet/luftkvalitet/ssen141812014/>
[Accessed: 2019-10-27].

Appendix A: Example Reports

Time: 2019-02-01 00:30:00	Sensor	Average	Percentage OK Signals	Values within Limits	Status
Time: 2019-02-01 01:00:00	Sensor	Average	Percentage OK Signals	Values within Limits	Status
Time: 2019-02-01 01:30:00	Sensor	Average	Percentage OK Signals	Values within Limits	Status
Time: 2019-02-01 02:00:00	Sensor	Average	Percentage OK Signals	Values within Limits	Status
Time: 2019-02-01 02:30:00	Sensor	Average	Percentage OK Signals	Values within Limits	Status
2019-02-01 02:30:00	CO	62	56	56	nonfunctional
2019-02-01 02:30:00	HCl	58	56	56	nonfunctional
2019-02-01 02:30:00	NOx	59	60	60	OK
2019-02-01 02:30:00	SOx	59	60	60	OK
2019-02-01 02:30:00	STOFT	56	52	52	nonfunctional
2019-02-01 02:30:00	TOC	61	56	56	nonfunctional
Time: 2019-02-01 03:00:00	Sensor	Average	Percentage OK Signals	Values within Limits	Status
Time: 2019-02-01 03:30:00	Sensor	Average	Percentage OK Signals	Values within Limits	Status
Time: 2019-02-01 04:00:00	Sensor	Average	Percentage OK Signals	Values within Limits	Status
Time: 2019-02-01 04:30:00	Sensor	Average	Percentage OK Signals	Values within Limits	Status
Time: 2019-02-01 05:00:00	Sensor	Average	Percentage OK Signals	Values within Limits	Status
Time: 2019-02-01 05:30:00	Sensor	Average	Percentage OK Signals	Values within Limits	Status

Report 1. Half hour summary. The red/orange rows indicate an error from the data and the cause is described in the status column.

	Överträdelser (Halvtimme): 2018-01					
	CO	HCl	NOx	SOx	STOFT	TOC
Överträdelser Gränsvärde 1 (97%)	0	0	0	0	0	0
Andel Godkänt	100.00	100	100	100	100	100
Överträdelser Gränsvärde 2 (100%)	0	0	0	0	0	0
Beräkningsgrundande halvtimmar	1363	1376	1373	1368	1370	1374
	Överträdelser (10 Min): 2018-01					
	CO					
Överträdelser Gränsvärde 1 (95%):	0					

Report 2. Month Report

Månadsrapport: 2018-01						
Dygnsmedelvärde						
	CO	HCl	NOx	SOx	STOFT	TOC
Gränsvärde	50	50	50	50	50	50
Enhet	ppm	ppm	ppm	ppm	ppm	ppm
Dygn						
1	104.11	101.72	102.33	105.33	107.82	101.72
2	102.01	91.76	87.23	90.23	104.95	87.23
3	FEL	105.6	FEL	FEL	111.93	104.24
4	60.04	60.3	50.91	56.42	61.02	57.75
5	90.18	84.21	84.26	87.26	89.98	84.26
6	92.96	79.78	90.13	65.96	84.59	62.96
7	63.99	57.89	61.37	63.95	60.01	60.42
8	96.72	94.68	97.84	99.56	97.82	98.09
9	FEL	FEL	FEL	FEL	FEL	FEL
10	99.73	62.74	89.03	FEL	102.35	FEL
11	143.24	140.24	140.24	148	148.65	140.24
12	121.44	119.29	113.45	119.39	125.55	FEL
13	59.04	61.04	57.57	64.04	64.7	61.04
14	57.1	51.16	FEL	60.09	61.2	53.02
15	61.01	FEL	55.23	58.23	59.37	56.01
16	FEL	101.46	94.74	97.74	111.87	FEL
17	FEL	43.09	45.62	81.57	88.71	78.57
18	111.28	106.59	99.89	103.79	112.13	105.12
19	FEL	80.21	75.56	FEL	81.89	75.56
20	110.99	103.61	106.14	102.68	109.2	101.31
21	99.81	99.12	96.1	99.1	102.62	95.57
22	FEL	FEL	FEL	FEL	FEL	FEL
23	53.06	50.06	53.04	55.79	53.06	50.06
24	73.79	67.5	70.79	73.79	75.03	69.54
25	65.44	64.43	60.35	63.35	68.29	61.74
26	106.2	117.84	102.53	112.01	120.06	108.69
27	44.51	43.11	39.44	46.11	44	43.11
28	69.58	61.41	67.01	67.48	65.09	60.34
29	112.98	107.04	FEL	FEL	113.46	103.85
30	80.81	84.19	79.12	86.1	92.36	80.74
31	86.37	FEL	87.08	86.37	92.54	82.7
Medelvärde	86.6556	82.96556	81.03846	83.7736	90.00862	80.14923

Report 2. Month Report

The month report provides a summary for the number of violations during the period. It also presents the daily mean values for the entire month and violations are marked orange and signal failures are marked with 'FEL'.



LUND
UNIVERSITY

Series of Bachelor's theses
Department of Electrical and Information Technology
LU/LTH-EIT 2019-737
<http://www.eit.lth.se>