YRKESHÖGSKOLAN NOVIA

Wear and Tear Statistics

Tools to analyse wear on engine components

Thomas Masus

Bachelor's Thesis

Information Technology

Vaasa 2011

BACHELOR'S THESIS

Author: Thomas Masus
Degree Programme: Information Technology
Supervisor: Kaj Wikman

Title: *Wear and tear statistics*

Date 26.4.2011 Number of pages 31

Summary

This Bachelor's thesis has been created for Technical Service W20 at Wärtsilä Finland Oy, Services. The task has been to collect, process and analyze wear data on different engine components. The goal with this project was to give all Wärtsilä personnel that needs updated data from measurement records about specific wear data, all needed information in the form of graphs. For this purpose a homepage was created in Wärtsilä's Intranet, Compass. Microsoft Excel has been used as the base for this project. The necessary information is stored in a database that delivers information through a web service. To be able to receive the information that the web service offers there had to be some coding done to the Excel document. The project will result in an Excel document that will be deployed on Compass.

Language: English Key words: Microsoft Excel, Microsoft SharePoint Server, Web Service

---

Filed at: Theseus.fi and The Tritonia Academic Library, Vaasa

OPINNÄYTETYÖ

Tekijä: Thomas Masus
Koulutusohjelma ja paikkakunta: Tietotekniikka, Vaasa
Ohjaajat: Kaj Wikman

Nimike: *Kulumistilastot*

Päivämäärä 26.4.2011 Sivumäärä 31

Tiivistelmä

Tämä opinnäytetyö on tehty Wärtsilä Finland Oy Servicesin Tekniselle Huollolle W20. Tehtävänä on ollut kerätä, käsitellä ja analysoida eri mootorikomponenttien kulumistietoja. Opinnääytetyön tavoite on ollut antaa kaikille Wärtsilän työtekijöille, jotka tarvitsevat tiettyjä ja päivitytettyjä mittaustietoja, tietoa graafisesti. Tätä varten järjestettiin tietty alue Wärtsilän intranettiin, Compassissa. Tälle projektille Microsoft Excel 2007 on ollut perusta. Tarvittava informaatio on varastoitu tietokantaan, joka lähettää informaation verkkopalvelun avulla. Informaatio, jonka verkkopalvelu lähettää on saatavilla, koska on käytetty ohjelmakoodeja Exceldokumenteissa. Työn takoituksena on ollut saada sellainen tulos, että Exceldokumentit on julkaistu Compassissa.

Kieli: englanti Avainsanat: Microsoft Excel, Microsoft SharePoint Serveri, verkkopalvelu

---

Arkistoidaan: Theseus.fi ja Tritonia, Vaasan tiedekirjasto

EXAMENSARBETE

Författare: Thomas Masus
Utbildningsprogram och ort: Informationsteknik, Vasa
Handledare: Kaj Wikman

Titel: *Slitagestatistik*

Datum 26.4.2011 Sidantal 31

Abstrakt

Detta ingenjörsarbete har gjorts åt Teknisk Service W20 vid Wärtsilä Finland Oy, Services. Arbetsuppgiften har bestått av att samla in, behandla och analysera förslitningsdata för olika motorkomponenter. Målet med detta ingenjörsarbete var att ge alla de personer inom Wärtsiläs koncern, vilka behöver ha uppdaterad data från mätprotokoll om specifik förslitningsdata, all behövlig information i form av grafer. För detta skapades ett hemområde på Wärtsiläs intranät, Compass. Det är Microsoft Excel som har använts som grund för projektet. Den behövliga informationen finns lagrad i en databas som levererar information med hjälp av en webbservice. För att kunna ta emot denna information som webbservicen levererar har programkod använts i Exceldokument. Detta arbete kommer att resultera i flera Exceldokument som kommer att publiceras på Compass.

Språk: engelska Nyckelord: Microsoft Excel, Microsoft SharePoint Server, webbservice

---

Förvaras: Theseus.fi och Tritonia, Vasa vetenskapliga bibliotek

TABLE OF CONTENTS

GLOSSARY

UDC

Universal Data Connection. A UDC file may contain information about several different types of data connections. The file extension is *.udcx or *.xml. The UDC file is defined to either query, submit or both. /21/

Data sources inside the UDC file are being stored on the Microsoft SharePoint Server so that they can be used by form templates and browser-enabled form templates. /9/

.NET

It is a platform for XML web services.

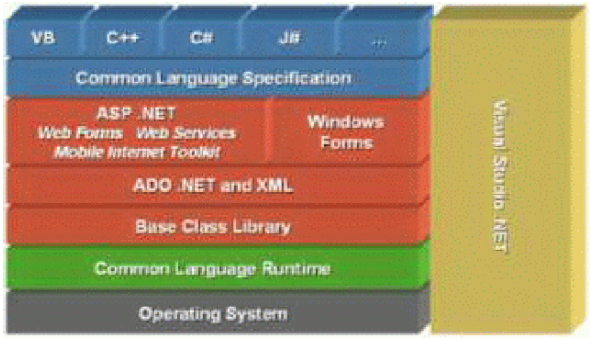


Figure 1.Architecture for .NET /25/

- *"Common Language Specification (CLS) – blue*
- *Framework Class Library (FCL) – red*
- *Common Language Runtime (CLR) – green*
- *.NET Tools – yellow"* /25/

Adobe Flex

Adobe Flex is created by Adobe Systems for development and deployment of cross-platform rich Internet applications based on the Adobe Flash platform. /1/

ASP.NET

Marketed by Microsoft and allows programmers to build dynamic web sites, web applications and web services. /2/

Compass

Wärtsilä had their intranet (internal web site) named Compass.

DataTable

The DataTable is a central object in the ADO.NET library. /7/

HTML

HyperText Markup Language. HTML is the basic building-blocks of Web pages. Information is being enclosed in angle brackets with opening tags and closing tags. This is simply a markup language used to tell a program how information is being displayed for the user. /10/

Live Meeting

Live meeting is a Microsoft Office product that allows users to participate in meetings held over the internet.

Microsoft Office InfoPath

Microsoft Office InfoPath is used for developing XML-based data entry forms. More information about how to use InfoPath can be found in section 3.3.5.

Microsoft Visual Studio C#

C# is a programming language with imperative, declarative, functional, generic, object-oriented (class-based), and component-oriented programming. /3/

Oracle Database

It is an object-relational database management system (ORDBMS). It is used to store data.

SOAP

Simple Object Access Protocol. SOAP transfers structured information through a Web Service. /18/

TDiK

Technical Data into Knowledge. It is a product which Wärtsilä developed in 2009 and it is their solution for how to handle measurement records. /19/

Web service

A web service is a method of communication between two electronic devices over a network. The interface is described in a machine-processable format called Web Services Description Language, WSDL. SOAP messages are used to interact with a web service. A web service is used to transact data. /22/

1 INTRODUCTION

This Bachelor thesis is made on behalf of Wärtsilä Finland Oy. My contact person was Niklas Donner, General Manager of Technical Services at Wärtsilä Finland Oy.

1.1 Wärtsilä Finland Oy

Wärtsilä Finland Oy is a company that produces both 4 stroke engines and 2 stroke engines. They are a leading supplier of engine solutions for both marine and power industries.

Figure 2. Wärtsilä's logotype /24/

Wärtsilä can be divided into three main groups: Ship Power, Power Plant and Services. Ship Power is the department where they focus on solutions at sea, while Power Plants focuses on electricity-producing engines on the mainland. Service, which is the biggest one of them all, takes care of the lifecycle of the engines. They offer the customers for example advice, spare parts and repairs.

The company has more than 17,000 employees in 160 different locations in 70 countries around the world. /24/

Wärtsilä key figures in 2010:

- *"Net sales EUR 4,553 million (5,260)*
- *Operating result EUR 487 million (638)*
- *Order intake EUR 4,005 million (3,291)*
- *Order book 31 Dec 2010 EUR 3,795 million (4,491)*
- *Personnel 17,528 (18,541)"* /24/

1.2 Technical Service

The Technical Service department is one of the many units in Wärtsilä Finland Oy located in Runsor, Vaasa. They are one of the most important technical knowledge centres within the services business of Wärtsilä supporting internal and external stakeholders technically during the life cycle of products and solutions delivered or serviced by Wärtsilä. /24/

In summer 2010 I worked as a summer trainee for the Technical Service W20 department. This is where I got this assignment to create a tool for wear and tear statistics. The W20-team is working with the following engine types: V14, W22, V24 and W20. The label V tells that the engine has been produced in Vaasa and the label W stands for Wärtsilä and does not specify where the engine is produced. The numerical part explains how big the inner diameter inside the cylinder liner is. The following organization chart shows the Wärtsilä Finland Technical Service organization set-up:

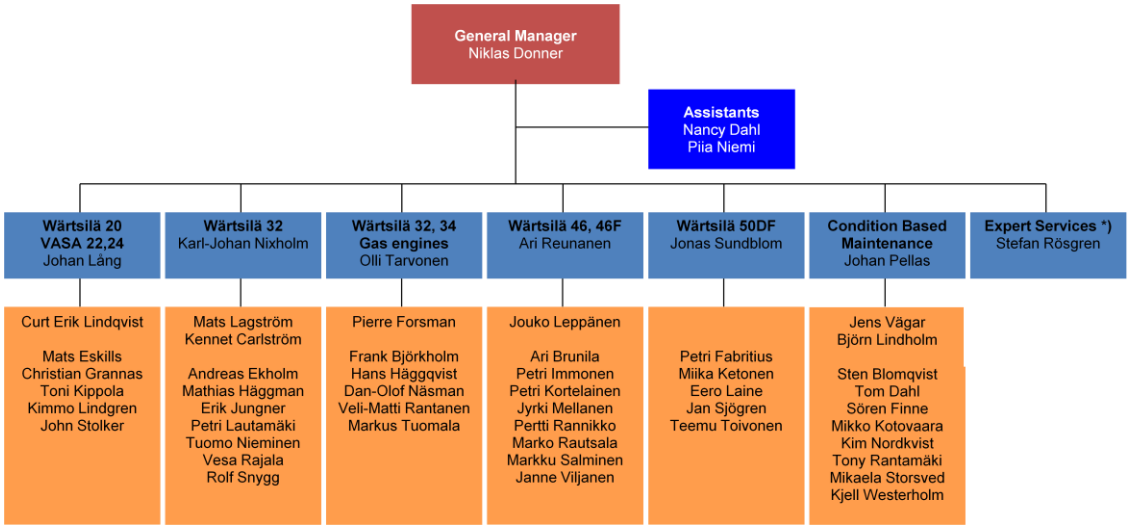


Figure 3. Organization set up. /20/

The main responsibilities within Technical Service are as follows:

- *"Make product improvements*
    - *Participation in the introduction of new engine types and other products*
    - *Make field follow up material*
    - *Participate in field tests*
    - *Support to warranty follow up*
    - *Make product improvements and up-grading/up-ratings*
    - *Make product Documentation*
    - *Initiate and write like: content of Service Bulletins, etc.*
    - *Give Code Resolution support*
- *Give Expert assistance like:*
    - *Give technical support for internal and external customer*
    - *Make site surveys for deeper engine and system analysis*
    - *Do trouble shooting in the field*
    - *Develop own measuring capability."* /20/

The Technical Service department consists for example of experts, superintendents, mechanical and electrical engineers and technicians.

1.3 Task

In 2009 Wärtsilä had a tool made called TDiK which stores measurement records. What it did not have was a way to present data visually with graphs.

Information data being stored in TDiK is very valuable, as it descibes the wear rate the engines have while they are in operation. When Wärtsilä is to set a warranty period for different components they have to base this warranty on some statistics. This is where the tool I have created would be handy.

Wärtsilä had TDiK built so that users could look at measurement records (MR) on a Web page containing tabular data. The user also had the possibility to extract chosen measurement record(s) to a Microsoft Excel document on his/her own computer.

TDiK contains measurement records of many components belonging to different engine sizes and types. The scope for “wear and tear statistics” was set to concern records for the W20 engine.

The first part of this assignment was to create a tool to analyze measurements. The result can be found in figures 11, 13 & 14. With this tool the following components were to be analyzed:

- Cylinder liner, wear rate
- Piston ring groove height, wear rate
- Connecting rod big end bearing bore, ovality

In the beginning it was necessary to learn what information the measurement records contained for each component, what information could explain wear rate and what affected the wear rate. The information that was needed for this analytical part is described in chapter 4.3.1.

The second part of the assignment was to connect these charts to a database so that the user could have a look at the latest measurement information. In the second part of the assignment it was said that the user should not have to read a manual to be able to get the charts made. To accomplish this some coding had to be done.

The last part was to make this tool available online to Wärtsilä personnel with the right privileges. To do this, the structures of TDiK and Microsoft SharePoint Server had to be studied to be able to transfer data between the two of them.

2 TDiK

The database TDiK is using is an Oracle database. The database is built so that the measurement records are categorized and stored as separate .XML files. The following picture illustrates the different modules of the system and processes between them:

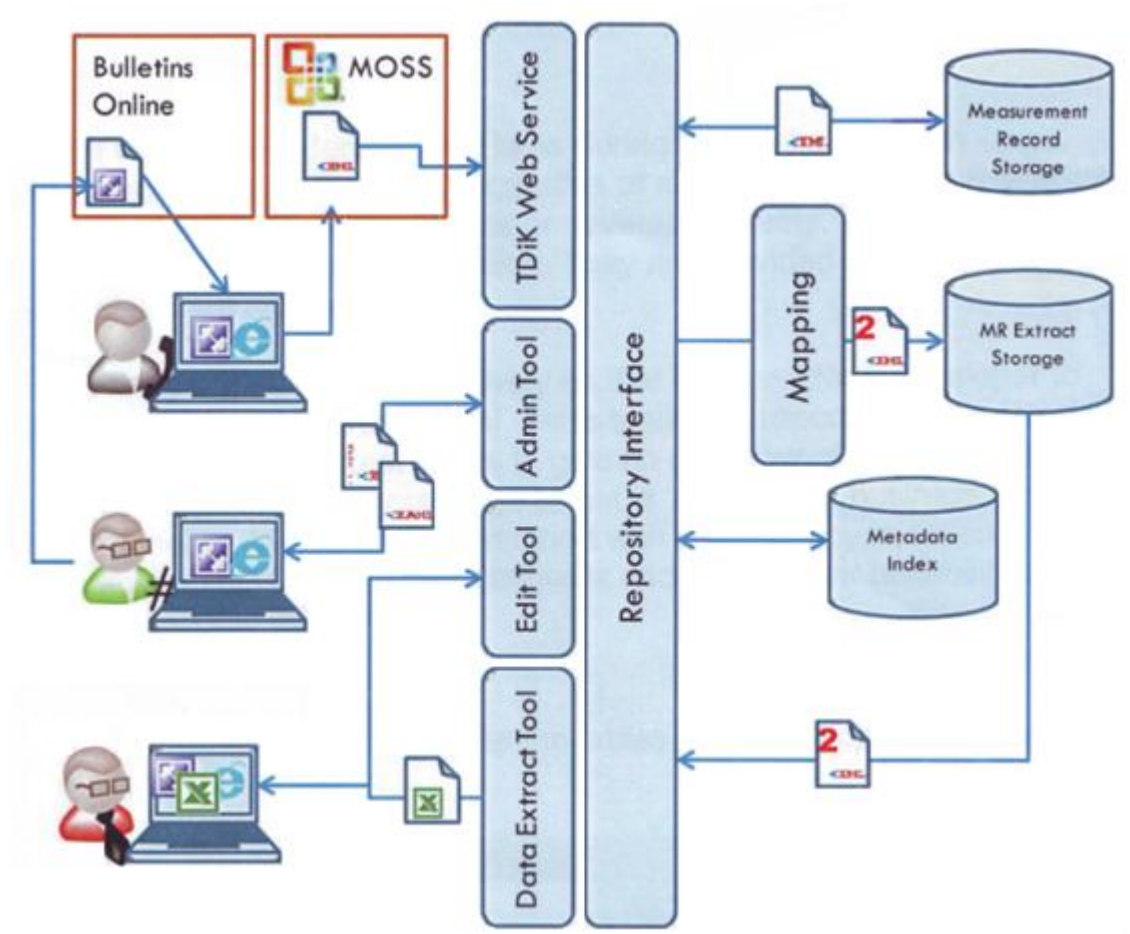


Figure 4. TDiK architecture. /19/

The individual services are designed with the Object Oriented (OO) approach using .NET and C# as practical tools. Non-OO platforms such as web pages, database and XML are designed from object to physical design. /19/

The procedure for how measurement records are being stored is as follows:

1. A field worker fills in a pre-defined InfoPath document (measurement record) and sends it away for approval.

2. A web service takes care of the XML file and sends it to a repository interface layer which indexes the measurement record and stores it. /19/

A web service (not mentioned in the previous picture) has been created separately for this project. By using SOAP calls, the web service will provide queried data from the TDiK server. The architecture for how this project is supposed to be used can be viewed in the figure below:

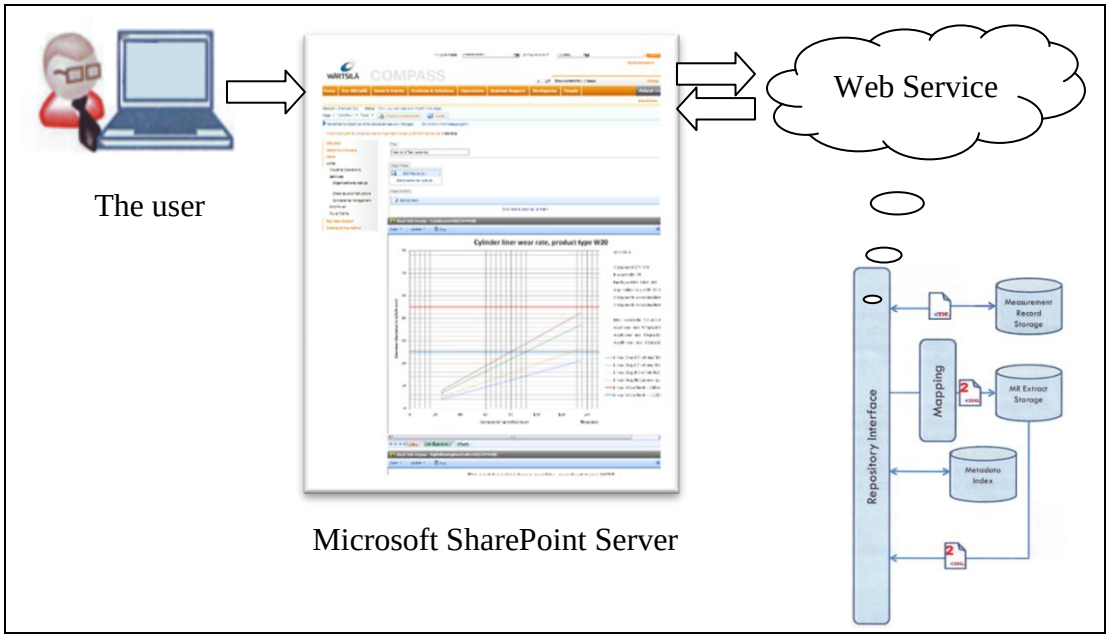


Figure 5. Architecture structure for this project

The format of which the web service provides data is in a single HTML tagged string. See the code example below:

```
<table border='1'>
<tr>
<th>Id</th>
<th>InstallationName</th>
<tr class="oddRow">
<td align="right">1234</td>
<td>theInstallationName</td>
</tr>
</table>
```

Code example 1.

3 TECHNOLOGIES

Wärtsilä in Switzerland has created a tool called “EB2S” using Adobe Flex 4. Its purpose was only to produce charts and display tabular data. It was built so that it is connected to many different data sources. This made it an interesting platform for “wear and tear statistics”. (Based on Live Meeting, 4.10.2010)

In the beginning this was the second alternative to Compass. After some more research on how TDiK provides data through its web service it was decided not to use it. The “EB2S” project is still in a development stage so trying to keep up with the development could lead to difficulties for the “wear and tear statistics” project.

Wärtsilä contains plenty of different applications with several connections all over the company, which is why this made it less interesting to have a connection from Switzerland to the TDiK database located in Finland.

3.1 Microsoft Office Excel 2007

Microsoft Office Excel 2007 is a tool used by many people. In Excel the user can create formulas and generate charts. An Excel spreadsheet can hold data of up to 1 million rows by 16 000 columns. /14/

The following figure shows the user interface for Microsoft Office Excel 2007:

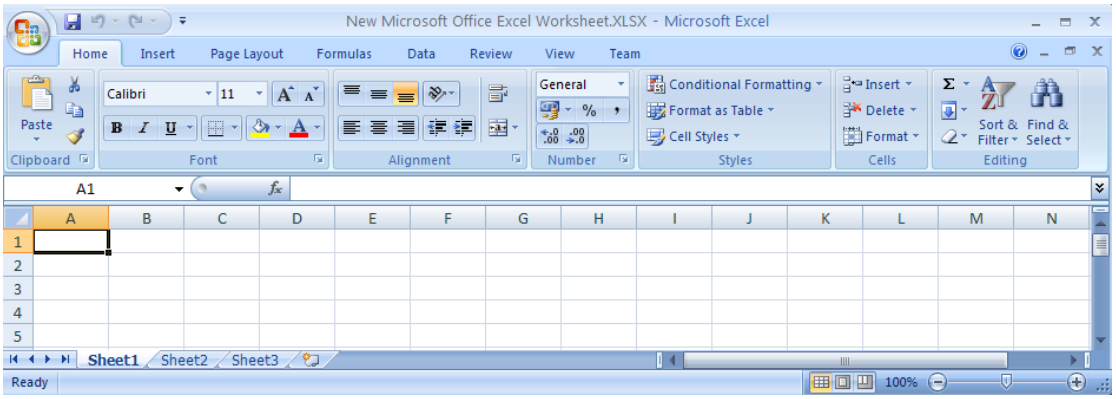


Figure 6. Microsoft Office Excel 2007 Workbook

Using Excel Services makes it possible to share and manage analyses with co-workers very easily. See chapter 3.3.4 – “Excel Services” for information about this. What this means is that Excel Services can dynamically render a spreadsheet as HTML, which makes it possible to access a spread sheet using a Web browser. See figure 9 on how to create a web part that supports excel workbooks.

Compressed Microsoft Office Excel XML format makes the document allocate less space, which means less storage consumption and bandwidth when transferred over internet. /14/

Microsoft Office Excel is a very commonly used tool within Wärtsilä. The workers who have their own company computer will get the Microsoft Office package (Excel, Word, InfoPath and many more) preinstalled when they are given a computer.

Excel gives the functionality to divide data into columns and rows, which makes it a useful tool for the user who will get a clearer view of the data. To create a chart it is simple to select the data the user wants to base his chart on and click “make chart”. There is so a lot of functionality built into Excel, and all cannot be mentioned here.

3.2 Visual Studio 2010 C# programming Excel 2007

With Visual Studio 2010 it is possible to create document-level and application-level add-ins for Microsoft Office Excel. These solutions can automate Excel, create more features in Excel and also modify the Excel user interface (UI). /16/

An example of how to automate Excel could be that charts can be programmatically created, worksheets can be formatted and cell values can be set.

Document-level customization has an assembly that will only be associated with a specific workbook. An example is shown in figure 7.

An application-level add-in has an assembly that is loaded into Excel. By using an add-in the assembly can be used by any Excel document.

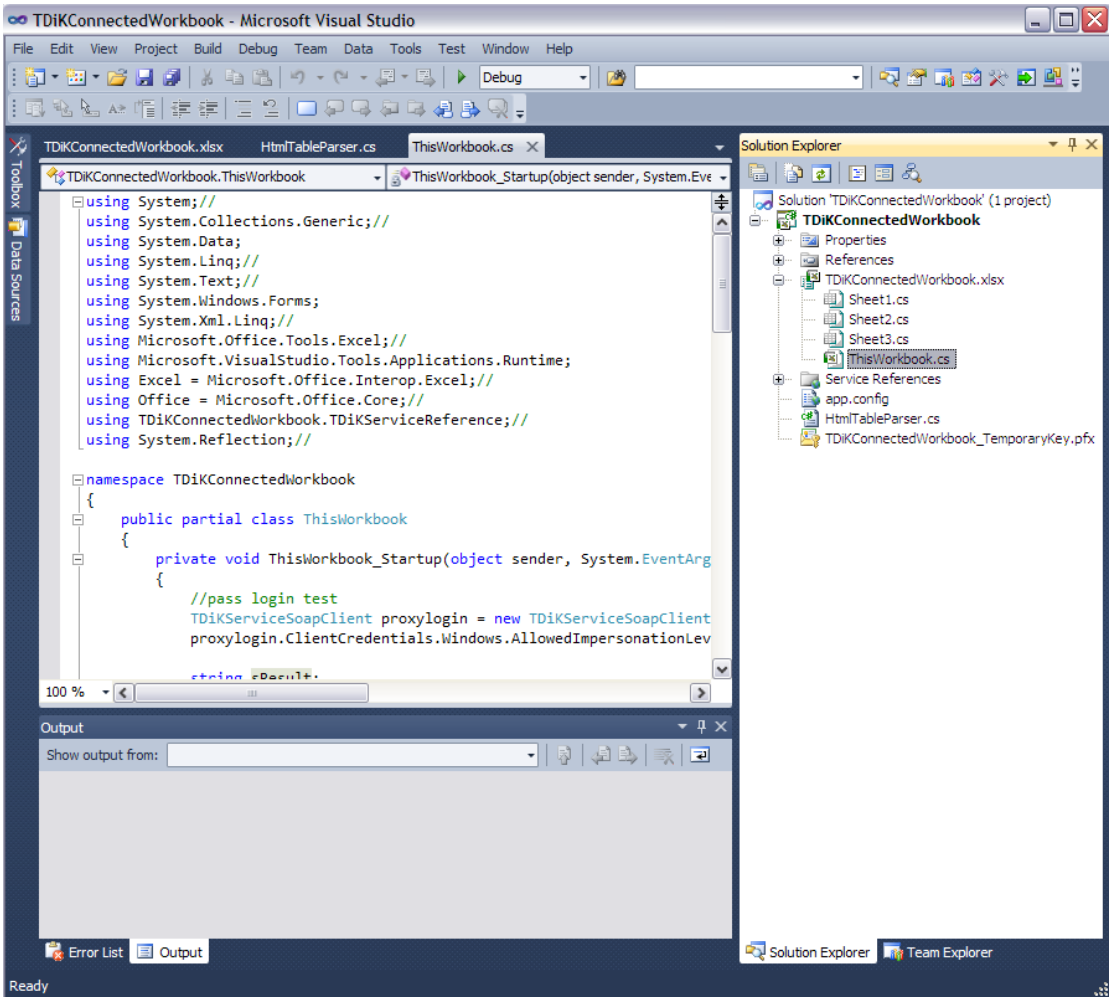


Figure 7. Microsoft Visual Studio programming Excel workbook

An alternative instead of programming an Excel document would have been to further develop TDiK by using Microsoft Visual Studio 2010 to create a chart on a web page using the C# programming language.

3.3 Microsoft SharePoint Server

Microsoft SharePoint Server is built on the ASP.NET framework. SharePoint Server operates with Microsoft Internet Information Services' (IIS) web server to create sites for collaboration, file sharing, web databases, social networking and web publishing. With SharePoint server it is possible to host web sites, portals, intranets, extranets and Internets. /15/

The following concepts which are parts of Microsoft SharePoint Server will be further explained in chapter 3.3.1 – 3.3.5.

3.3.1 Compass

Wärtsilä had their intranet built with Microsoft SharePoint Server 2007 and it was given the name Compass. The personnel in Wärtsilä can get the latest information about the company through this Web site. The following figure shows the user interface:

Figure 8. Compass user interface /6/

3.3.2 Lists

In Microsoft SharePoint Server it is possible to create lists. A list is used to store the same sorts of data that you would normally place into a Microsoft Office Excel spreadsheet. There are exceptions to the rule though. For instance, you can attach documents to list entries. /17/

Lists can be useful because they can receive data from web parts and pass the data to another web part.

3.3.3 Web part

A Microsoft SharePoint Server page contains regions and inside those regions there are web part zones. Web parts are reusable and it is easy to configure their content to present the information in the most beneficial way. Figure 9 shows an example.

Web parts can be configured to receive external data from databases or file systems.

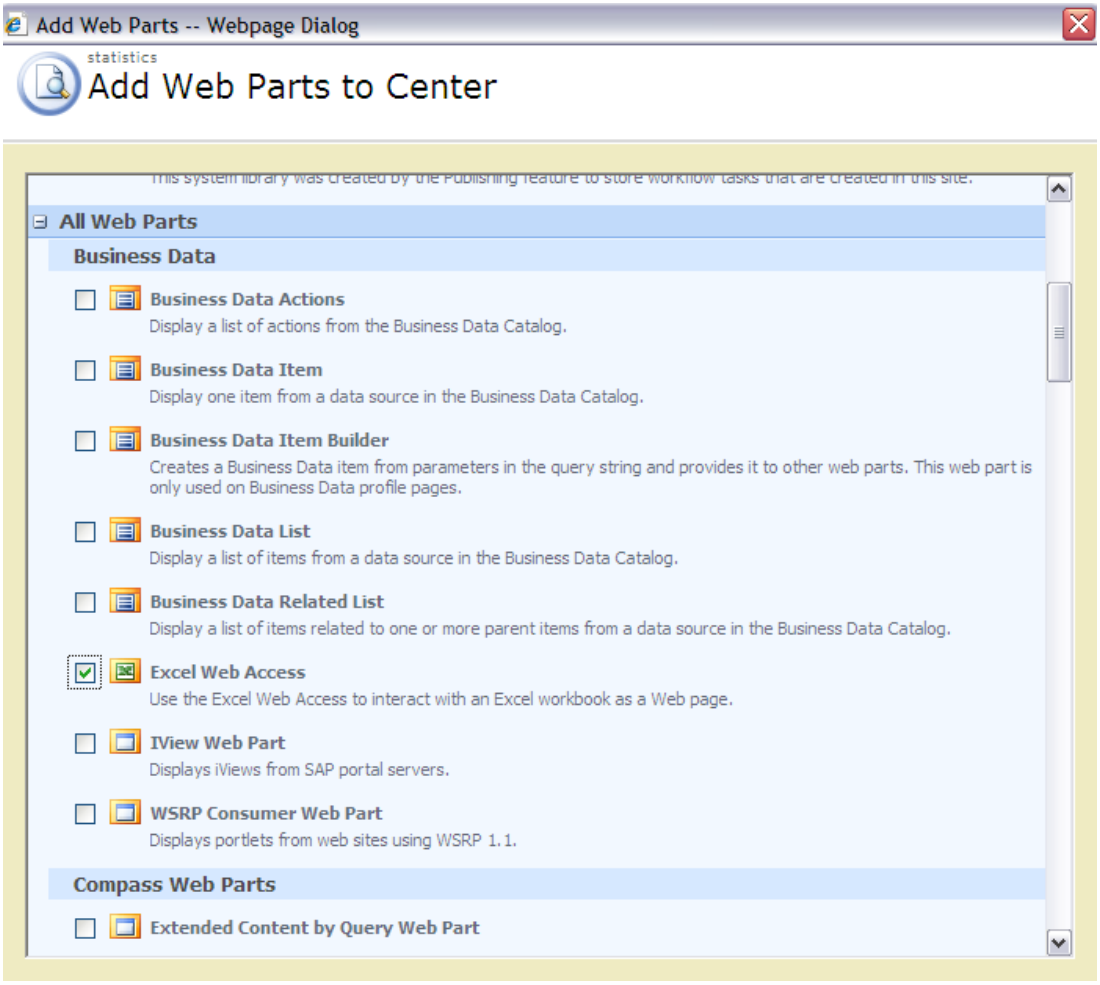


Figure 9. How to add Web Parts in Microsoft SharePoint Server

3.3.4 Excel Services

Excel Services is a service application to load, calculate, and display Microsoft Excel workbooks on Microsoft SharePoint Server. The way to use Excel Service is to choose to publish to a location on a Microsoft SharePoint Server site. See the example in figure 10. After that it is possible to open this document inside a web part. This Excel

workbook can be connected to a Microsoft SharePoint List, mentioned in chapter 3.3.2, which makes it possible for a worksheet to have it as a data source. The figure below shows what options the user has when uploading an Excel file. /8/

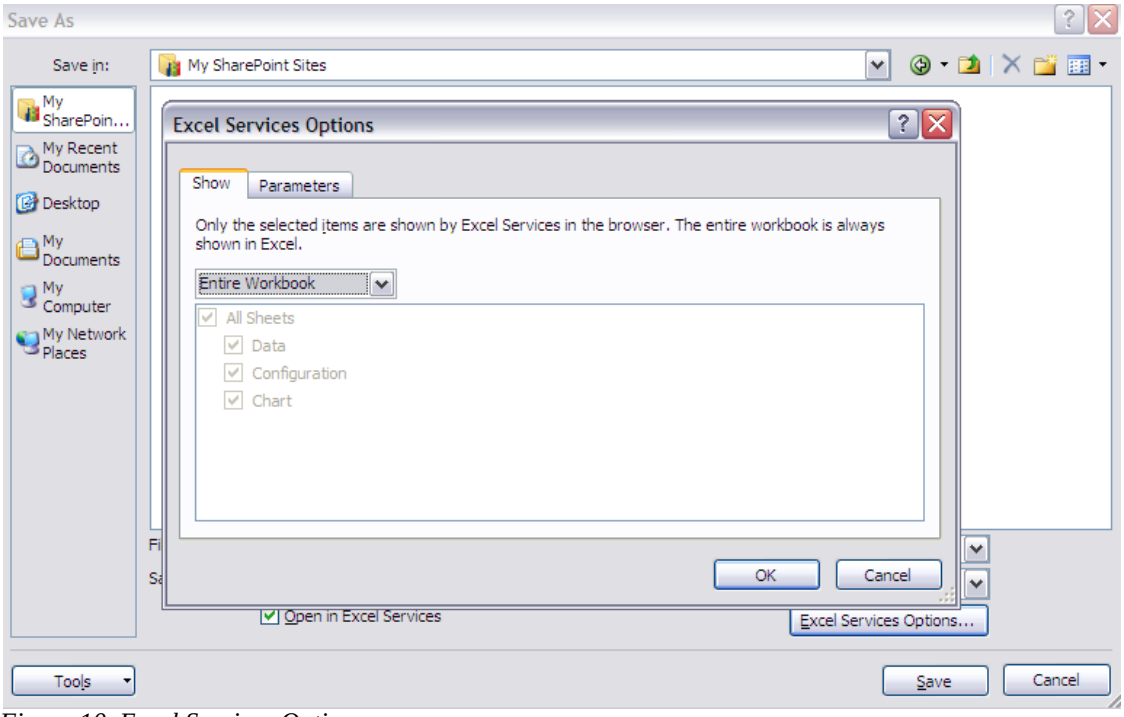


Figure 10. Excel Services Options

This becomes very useful when workbooks or worksheets are to be uploaded and shown inside a web part. The Excel document will still have many of its functionalities on the web. There are some features that it does not support. For more information, see table 1 below:

Table 1. Unsupported features in Excel services. /9/

FEATURE	COMMENTS
VBA	*Visual Basic for Applications (VBA) code, macros, add-ins, and user defined functions (UDFs).* *A programmer can customize Excel Services in many ways, including the creation of a user-defined function (UDF). For more information, see the Microsoft Office SharePoint Server 2007 Software Development Kit (SDK).*
Legacy macro languages	*Microsoft Excel 4.0 Macro Functions and Microsoft 5.0 dialog sheets.*
Controls	*Form toolbar controls, Toolbox controls, and all ActiveX controls.*
XML	*XML maps and embedded smart tags.*
Security and privacy	*Workbooks, worksheets, or ranges with protection, and workbooks that have Information Rights Management (IRM).* *To protect workbooks in Excel Services, use Microsoft Windows SharePoint Services rights and permissions.*
Images and objects	*Linked or embedded objects or images, inserted pictures, AutoShapes, WordArt, and diagrams, such as organization charts.*
Ink	*All ink features including drawing, writing, and annotations.*
OLE and DDE	*Object Linking and Embedding (OLE) objects and Dynamic Data Exchange (DDE) links.*
Displayed formulas	*Workbooks saved with formulas that are displayed.*
Data validation	*Preventing invalid data entry and creating drop-down lists.*
Data sources	*Data retrieval services for Microsoft Business Solutions, Windows SharePoint Services lists, Microsoft SQL Server, external data ranges (also called query tables), and tables linked to Windows SharePoint Services lists.*
Queries	*Web queries and text queries.*
External references to linked workbooks	*Creating external references (also called links) to a specific cell range, to a defined name for the specific cell range, or as part of a name definition.*
Comments	*Display and adjustment of comments.*
Consolidation	*Consolidated data in PivotTable reports.*
Shared workbooks	*Sharing of workbooks and resolving conflicting changes.*
Digital signatures	*Visible and invisible digital signatures in a workbook.*
Attached toolbars	*Custom toolbars attached to the workbook by using Office Excel 2003 before the workbook was converted to Excel 2007.*

3.3.5 InfoPath Form Services

InfoPath Form Services is a part of Microsoft SharePoint Server. When it is used on the Microsoft SharePoint Server the user can browse InfoPath documents without having InfoPath installed on his/her computer. /13/

By using InfoPath Form Services the user will have the possibility to fill in a form to either post or query information inside a web part on a Microsoft SharePoint Server site. The InfoPath document can post data into a Microsoft SharePoint List.

4 IMPLEMENTATION

In the beginning the plans were to create an InfoPath document inside a web part that would work as a User Interface (UI) for the user to fetch data from TDiK's web service. It would send data to a Microsoft SharePoint List which would provide a specified Excel worksheet with measurement data.

This project has not come that far yet so the user will only be provided with the latest data from the web service using an Excel workbook on his own computer. The user will still have all the functionality to filter data that has been given and will be able to view a chart.

4.1 Deployment

Wärtsilä already has an intranet built with Microsoft SharePoint Server 2007 called Compass. When we chose to use Compass we got another thing for free: the user has to be inside the local network and have a Wärtsilä login account on Compass to be able to see the material. The following picture gives a general overview of what settings can be configured to restrict the appearance for the user:

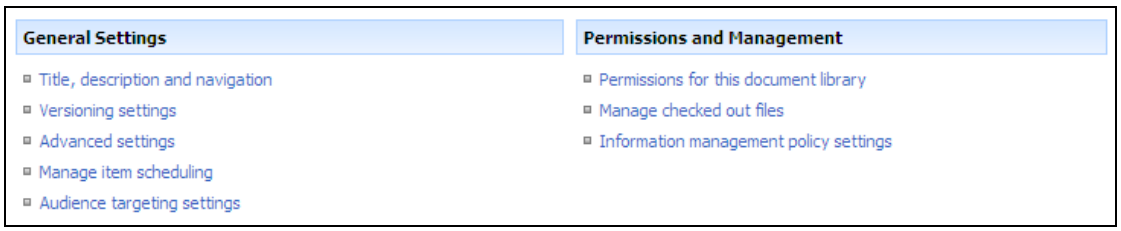


Figure 11. Site configuration

A Web page was created called "wear and tear statistics" which will contain the analytical tools that give the user information on the wear rate of different components. The Web page can be viewed in figure 18.

4.2 Database

All data needed for this project is stored in TDiK's Oracle database. The way to reach the data was through a web service belonging to TDiK. A test web service was set up just for the purpose of analyzing data. At the moment the web service is only providing data from the development area (a sandbox where tests can be made). This will be changed and the production database will be used as the source later on so the data will be the real data coming from field works.

4.3 Excel development

Microsoft Excel 2007 was chosen as the tool to do all the calculations and create charts. This is because Excel fulfills all needed functionality and Microsoft SharePoint Server supports Excel very well.

Three worksheets had to be created: Data, Configuration & Chart. Worksheet "Data" will contain data from the measurement records. Worksheet "Configuration" will handle calculations and define labels in the worksheet "Chart". The worksheet named "Chart" will contain information from both worksheets "Data" and "Configuration".

Figures 12, 14 and 15 show the cylinder liner workbook sheets. The same workbook look has been created to analyze the connecting rod big end bearing bore and the piston ring groove height.

4.3.1 Worksheet "Data"

This worksheet will contain all the raw data that the web service provides. The columns will contain:

ProductRefType, ProductNumber, MeasurementDate, CountEngine (which is a concationation of Productnumber and MeasurementDate; this column is used to count how many unique engines there are), PartOperatingHour, OperatingHour, Application, FuelType, Max_1, Average_I, Average_II, Average_III, NewPartOperatingHour (is a

new column and will get the value from PartOperatingHour and if no PartOperatingHour is set then the value will be taken from OperatingHour).

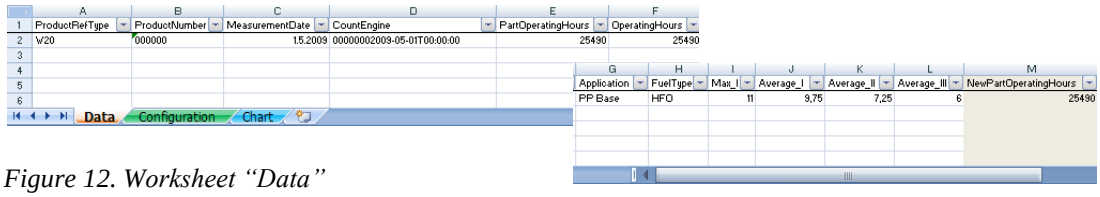


Figure 12. Worksheet "Data"

Name ranges have been used here so that all columns have their own name and depending on how many rows are being used the name range stops there. This has been created using Microsoft Excel Name Manager. Excel Services supports name ranges, which means that they will also work on Microsoft SharePoint Server.

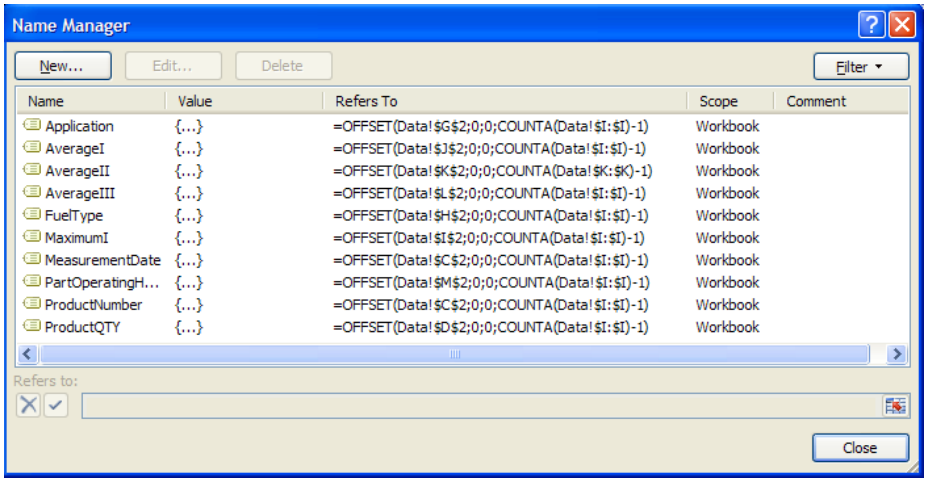


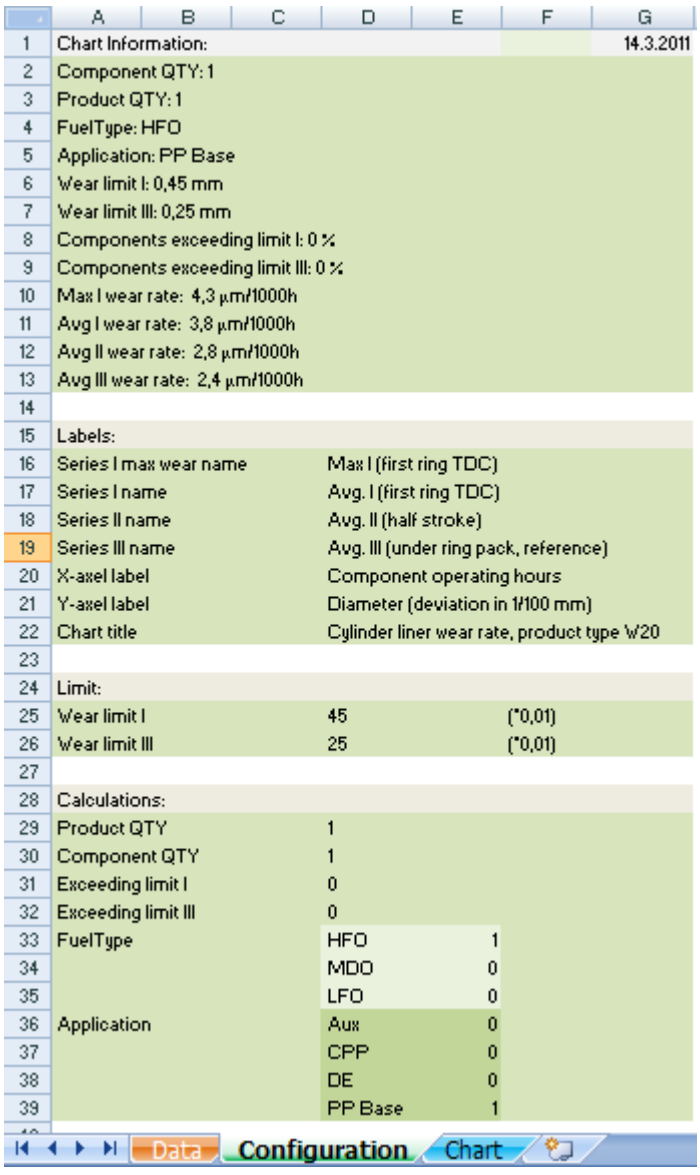
Figure 13. Name Manager for worksheet "Data"

4.3.2 Worksheet "Configuration"

This worksheet will make all necessary calculations. Since all data in worksheet "Data" has name ranges, the calculation part became easier since the dynamical amount of rows does not have to be taken into account. In this sheet the user can also have a good overview of how many components are running on different fuel types and applications. This is shown in figure 14.

Functions used in the “Configuration” worksheet were:

- Concatenate()
- If()
- Sumproduct()
- Counta()
- Countif()
- Linest()
- Today()



	A	B	C	D	E	F	G
1	Chart Information:						14.3.2011
2	Component QTY: 1						
3	Product QTY: 1						
4	FuelType: HFO						
5	Application: PP Base						
6	Wear limit I: 0,45 mm						
7	Wear limit III: 0,25 mm						
8	Components exceeding limit I: 0 %						
9	Components exceeding limit III: 0 %						
10	Max I wear rate: 4,3 µm/1000h						
11	Avg I wear rate: 3,8 µm/1000h						
12	Avg II wear rate: 2,8 µm/1000h						
13	Avg III wear rate: 2,4 µm/1000h						
14							
15	Labels:						
16	Series I max wear name			Max I (first ring TDC)			
17	Series I name			Avg. I (first ring TDC)			
18	Series II name			Avg. II (half stroke)			
19	Series III name			Avg. III (under ring pack, reference)			
20	X-axel label			Component operating hours			
21	Y-axel label			Diameter (deviation in 1/100 mm)			
22	Chart title			Cylinder liner wear rate, product type W20			
23							
24	Limit:						
25	Wear limit I			45		("0,01)	
26	Wear limit III			25		("0,01)	
27							
28	Calculations:						
29	Product QTY			1			
30	Component QTY			1			
31	Exceeding limit I			0			
32	Exceeding limit III			0			
33	FuelType			HFO	1		
34				MDO	0		
35				LFO	0		
36	Application			Aux	0		
37				CPP	0		
38				DE	0		
39				PP Base	1		

Data | Configuration | Chart

Figure 14. Worksheet “Configuration”

4.3.3 Worksheet "Chart"

The third worksheet contains the chart which explains the wear rate graphically. This chart is shown in figure 15. The information did not have to be more thoroughly explained inside the charts since the user will already have some background knowledge about engines. The information inside the chart is based on data from both the "Data" and the "Configuration" worksheets. All measuring points are taken from the "Data" worksheet and plotted in the chart as dots and all information about the dots is taken from the "Configuration" sheet.

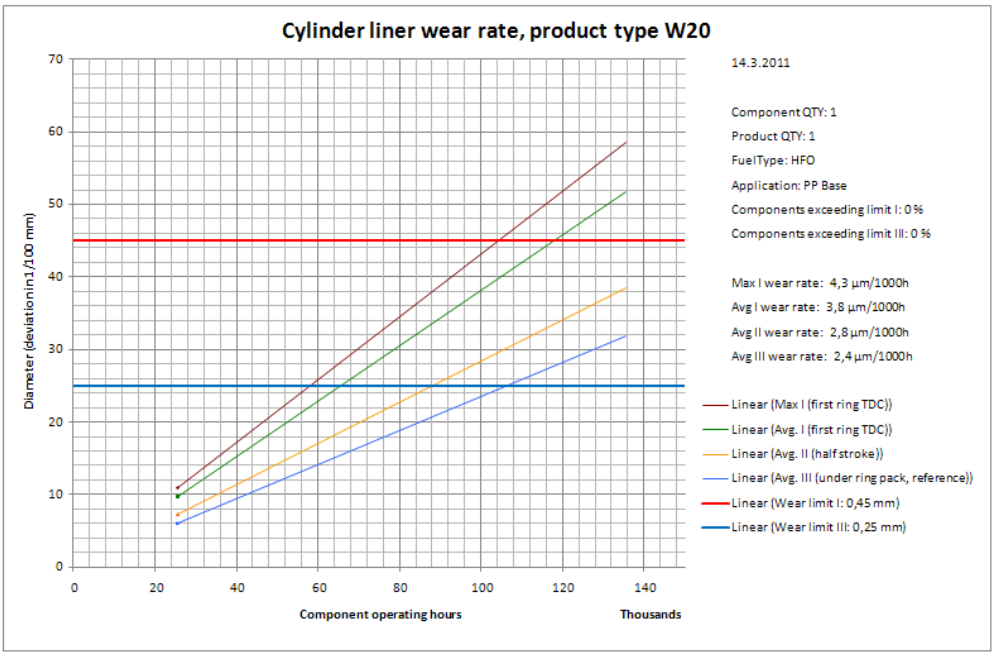


Figure 15. Worksheet "Chart"

The trend lines are linear based on each data series. They have their intercept beginning at zero, this because there is no wear when a component is new. The more measuring points that are taken into account, the more reliable the trend lines will become.

4.4 Data connections

What is already made is an Excel 2007 workbook containing dynamically defined name ranges and calculation formulas. To obtain “live data” some programming had to be done. By using Microsoft Visual Studio 2010 it is possible to open a workbook and insert code into it. This is shown in figure 17.

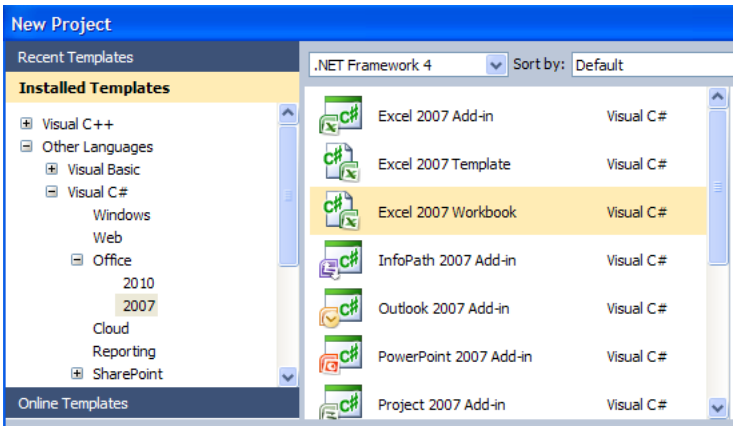


Figure 16. How to create Excel 2007 Workbook Project

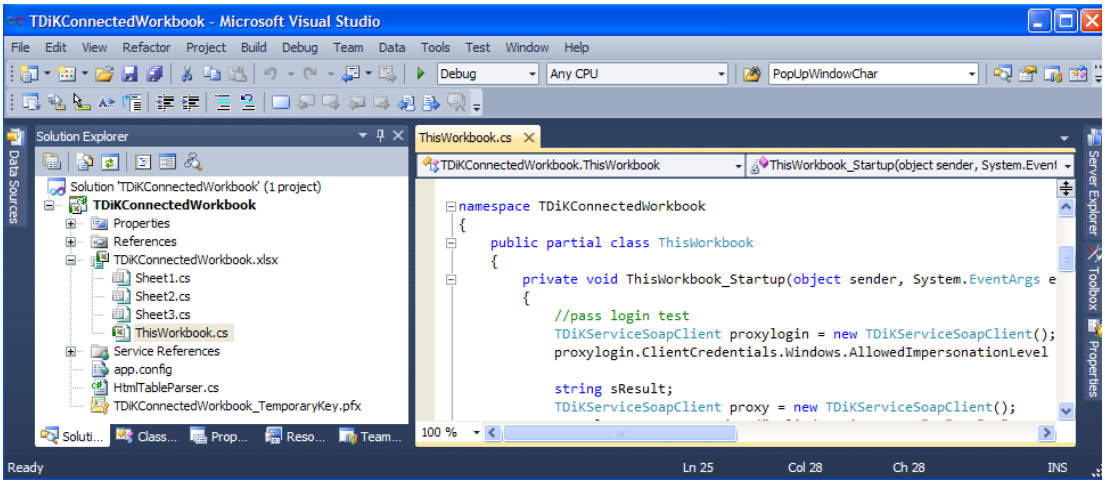


Figure 17. Programming Excel 2007 Workbook Project

4.4.1 Web service connection

A Service Reference to the TDiK Web Server address was added. The return collection type was set to be a "Generic List". The Workbook will now know how to communicate with the web service. The web service has a function "GetMRdata", which has four input parameters and will return queried data. The following code is being run when the Excel workbook is opened by the user:

Code example 2 shows how the web service is being SOAP called:

```
string sResult;
TDiKServiceSoapClient proxy = new TDiKServiceSoapClient();
sResult = proxy.GetMRdata(”parameter1”, "Parameter2", "Parameter3",
"Parameter4");
```

Code example 2.

The queried data when calling the web service returned a single text string which contains all the information. The information inside the string was HTML-tagged data, which means that the only way to be able to tabulate this data into Excel was to create an HTML-parser (create some form of internal representation). I found a parser that would convert HTML into DataTable. /11/

The code for the parser was implemented as a class inside the project named "HtmlTableParser.cs". An easier explanation of what it does is that it generates headers and rows and match them with each other. The code below describes the solution for how the parser works.

This code is used inside the class to tell the program how to determine what is tables, headers, rows and cells.

```
private const string CommentPattern = "<!--(.*?)-->";
private const string TablePattern = "<table[^>]*>(.*?)</table>";
private const string HeaderPattern = "<th[^>]*>(.*?)</th>";
private const string RowPattern = "<tr[^>]*>(.*?)</tr>";
private const string CellPattern = "<td[^>]*>(.*?)</td>";
```

Code example 3. /11/

The “ParseTable” takes the HTML string and parses it into a DataTable:

```
public static DataTable ParseTable(string tableHtml)
{
string tableHtmlWithoutComments = WithoutComments(tableHtml);

DataTable dataTable = new DataTable();

MatchCollection rowMatches = Regex.Matches(
tableHtmlWithoutComments,
RowPattern,
ExpressionOptions);

dataTable.Columns.AddRange(tableHtmlWithoutComments.Contains("<th")
                                          ? ParseColumns(tableHtml)
                                          : GenerateColumns(rowMatches));

ParseRows(rowMatches, dataTable);

return dataTable;
}
```

Code example 4. ParseTable /11/

If the string contains any form of comments, this will remove those parts:

```
private static string WithoutComments(string html)
{
return Regex.Replace(html, CommentPattern, string.Empty, ExpressionOptions);
}
```

Code example 5. /11/

Add rows to DataTable and match with input MatchCollection:

```
private static void ParseRows(MatchCollection rowMatches, DataTable
dataTable)
{
foreach (Match rowMatch in rowMatches)
{
// Do this as long as it is not a header
if (!rowMatch.Value.Contains("<th"))
{
DataRow dataRow = dataTable.NewRow();

MatchCollection cellMatches = Regex.Matches(
rowMatch.Value,
CellPattern,
ExpressionOptions);

for (int columnIndex = 0; columnIndex < cellMatches.Count; columnIndex++)
{
dataRow[columnIndex] = cellMatches[columnIndex].Groups[1].ToString();
}

dataTable.Rows.Add(dataRow);
}
}
}
```

Code example 6. /11/

DataColumns defines what will be the columns in the DataTable:

```
private static DataColumn[] ParseColumns(string tableHtml)
{
MatchCollection headerMatches = Regex.Matches(
tableHtml,
HeaderPattern,
ExpressionOptions);

return (from Match headerMatch in headerMatches
select new DataColumn(headerMatch.Groups[1].ToString())).ToArray();
}
```

Code example 7. /11/

If the string coming from the web service does not have a specified header cell then by counting how many numbers of cells there are in a row the DataColumn can be calculated (Assuming all rows have the same amount of cells).

```
private static DataColumn[] GenerateColumns(MatchCollection rowMatches)
{
   int columnCount = Regex.Matches(rowMatches[0].ToString(),CellPattern,
   ExpressionOptions).Count;

return (from index in Enumerable.Range(0, columnCount)
select new DataColumn("Column " + Convert.ToString(index))).ToArray();
}
```

Code example 8. /11/

There are different ways to program a Microsoft Excel Workbook. One option is to insert code inside a specific worksheet or, as I did, to place it inside "ThisWorkbook.cs" (referring to figure 17). The code is programmed so that when the document is opened the code will be run. This will make it easier for the user. He will not have to know how to set up the connection.

When trying to call the web service I discovered that guests/anonymous could not SOAP call the TDiK web service. The user has to send his login information inside Wärtsilä. This was handled by the following code:

```
TDiKServiceSoapClient proxylogin = new TDiKServiceSoapClient()
proxylogin.ClientCredentials.Windows.AllowedImpersonationLevel =
System.Security.Principal.TokenImpersonationLevel.Impersonation;
```

Code example 9. C#-code inside ThisWorkbook.cs. /12/

Also the security mode had to be modified. Inside app.config the following modifications were set:

```
<security mode="TransportCredentialOnly">
<transport clientCredentialType="Windows"
           proxyCredentialType="None"
           realm="" />
<message clientCredentialType="UserName" algorithmSuite="Default" />
</security>
```

Code example 10. Security mode configuration inside app.config. /4/

4.4.2 Worksheet programming

When parsing incoming data into a data table it had to be somehow inserted into the worksheet “Data”. The following code is run after the SOAP call has been made. What it does is to iterate through the DataTable and tabulate data into the Excel worksheet “Data”:

```
Excel.Workbook oWB;
Excel.Worksheet oSheet;

// Define active workbook
oWB = Globals.ThisWorkbook.Application.ActiveWorkbook;

oSheet = oWB.Sheets["Data"];
oSheet = oWB.ActiveSheet;

// Process the DataTable
DataTable dt = HtmlTableParser.ParseTable(sResult);

int rowCount = 1;
foreach (DataRow dr in dt.Rows)
{
rowCount += 1;
for (int i = 1; i < dt.Columns.Count + 1; i++)
{
// Insert Column names
if (rowCount == 2)
{
oSheet.Cells[1, i] = dt.Columns[i - 1].ColumnName;
}
oSheet.Cells[rowCount, i] = dr[i - 1].ToString();
}
}
```

Code example 11. How to insert data into the worksheet ”Data”. /23/

5 RESULT AND DISCUSSION

5.1 Result

What I have created is a tool to analyze the wear rate of W20 engine components. The user can filter chosen data that he or she wants to have more information about. The graph explains to the user how much wear the component has per running hour. This graph can be used for comparing what impact different fuel types can have on a component's wear rate.

It is also possible to filter data according to which application that is being used. This data explains if the engine is used for example in a power plant, as a main engine in a vessel/boat or as a help-engine.

A web page where these tools will be located has been created. The user will find this tool internally by browsing to the web page "wear and tear statistics".

The Excel workbooks have not yet been uploaded to Compass. They have only been tested locally on my computer to receive data from the web service. There is a possibility to upload the Workbook to Microsoft SharePoint Server as it is and the users will have to download it to their computer for distribution.

What I have accomplished with this project is to research what parts are needed to connect a simply made web service to Microsoft SharePoint Server. The Excel workbook handles incoming data from an analytical point of view and presents it to the user.

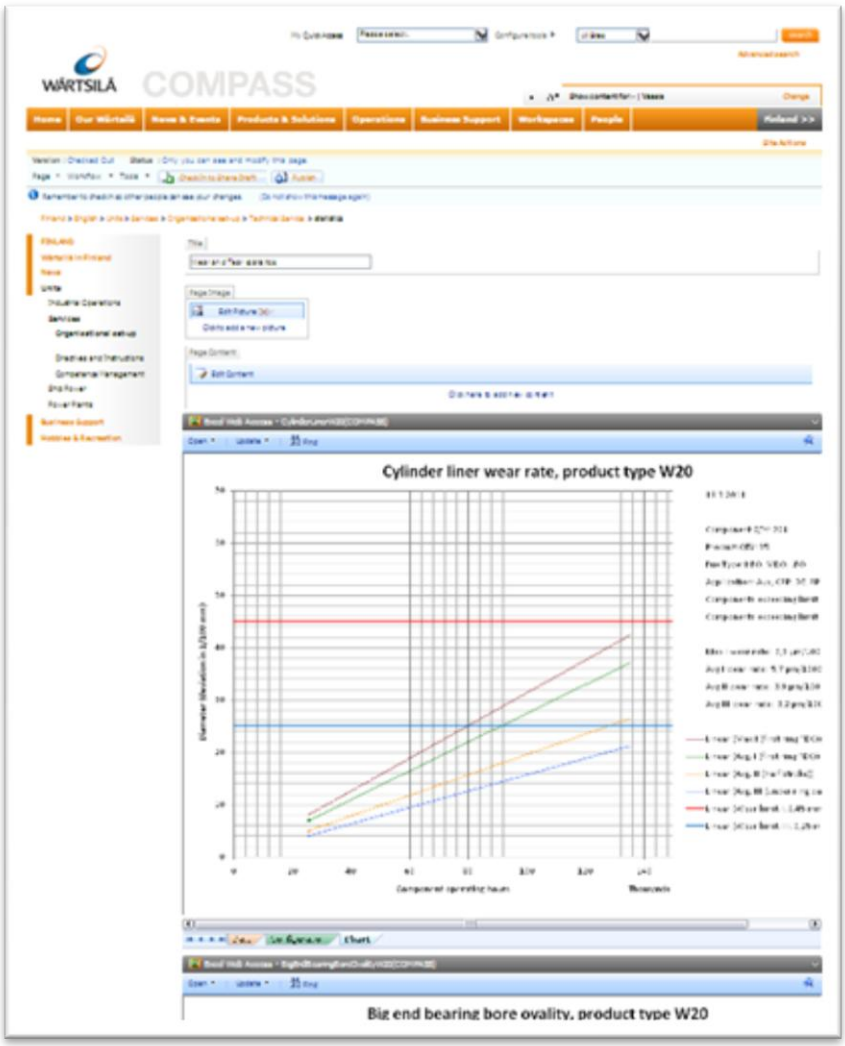


Figure 18. A view of what the "wear and tear statistics" homepage looks like.

Figure 18 gives a hint of what the end result will look like. As the figure is showing this is possible to make by using static data inside the workbook and upload it using Excel Services and open it in a Web Part. Now in the beginning this could be a short-term solution for analyzing wear statistics.

5.2 Further development

The next step is to tell Compass to allow the connection between the UDC file, the web service and the Excel workbooks.

The handling of transfer errors between the Workbook and the web service is one of the things that could be developed further. Error messages would be another aspect to look into. These could be handled with code using Microsoft Visual Studio 2010.

The Compass Web page “wear and tear statistics” only has the basic structure for now. There has to be some further development of what the user interface should look like.

These tools could be implemented to analyze all engine types and their components since all have quite the same information inside their measurement records.

These things have been taken into account but there was not enough time to develop them further more.

5.3 Discussion

This was a very interesting and educational project since many pieces had to be thought of and fall into place to make this project possible. I had to learn about Wärtsilä and their products and how to set up meetings. I also learned more about Microsoft software.

As Technical Service is no IT department I had to set up meetings using Live Meeting with people within the company and also arrange group meetings. A Live Meeting is held over an Internet connection whereas arranging group meetings means that all participants have to have time in their schedule to attend the meeting, a room has to be available and the agenda has to be made so that participants will know what topics will be discussed during the meeting. Since people can have many meetings in one day it is necessary to define when and where the meeting will be held and for how long it will last.

The main obstacle in this project was time itself. Almost all communication had to be done by email and this made the process very time-consuming. But instead of letting the time go by I had time to dig more into what software and programs are being used within Wärtsilä.

The good thing with working with computers is that you are not the only one having difficulties, there are many more people who have the same kind of questions. A lot of time was spent on reading forums and home pages to understand and learn about solutions and technologies.

The software and technologies I had to learn and understand were:

- Microsoft Office 2007 (InfoPath, Excel)
- Microsoft SharePoint Server 2007 (Web Parts, Lists, Services)
- What is a UDC file and how to use it
- What is a web service and how can it be used
- What is SOAP
- Parse a HTML-tagged string

This is the minimum amount of components that was needed to be able to create this project solution.

From this project I have learned how to set up a small project and be the one in charge of creating a solution for the “customer”.

I am very satisfied with this project and how it turned out. Further development is necessary to make this project usable for the whole company, which is the intention in the near future.

6 LIST OF REFERENCES

1. Adobe Flex (24.3.2011)
   http://en.wikipedia.org/wiki/Adobe_Flex
   (Read 18.3.2011)
2. ASP.NET (3.4.2011)
   http://en.wikipedia.org/wiki/ASP.NET
   (Read 18.3.2011)
3. C Sharp programming language (1.4.2011)
   http://en.wikipedia.org/wiki/C_Sharp_(programming_language)
   (Read 10.2.2011)
4. Client authentication (20.5.2010)
   http://stackoverflow.com/questions/1044034/wcftestclient-the-http-request-is-unauthorized-with-client-authentication-scheme
   (Read 16.3.2011)
5. Compass, Wärtsilä intranet (2010)
   http://www.simply-communicate.com/case-studies/company-profile/wartsila-points-its-compass-right-direction
   (Read 27.2.2011)
6. Create and Use a Data Connection Library (2010)
   http://msdn.microsoft.com/en-us/library/ms772101.aspx
   (Read 4.1.2011)
7. DataTable (11.3.2011)
   http://msdn.microsoft.com/en-us/library/system.data.datatable.aspx
   (Read 18.3.2011)
8. Excel Services (n.d)
   http://msdn.microsoft.com/en-us/library/ms546696.aspx
   (Read 20.10.2010)
9. Excel Services supported and unsupported features (n.d)
   http://office.microsoft.com/en-us/excel-help/differences-between-using-a-workbook-in-excel-and-excel-services-HA010021716.aspx
   (Read 20.10.2010)

10. HTML (4.4.2011)
http://en.wikipedia.org/wiki/HTML
(Read 16.3.2011)
11. Html table parser (6.5.2010)
http://blog.hypercomplex.co.uk/index.php/2010/05/parsing-html-tables-into-system-data-datatable/
(Read 9.3.2011)
12. Impersonating the Client (n.d)
http://msdn.microsoft.com/en-us/library/ms751513.aspx
(Read 22.3.2011)
13. InfoPath Forms Services Overview (2010)
http://msdn.microsoft.com/en-us/library/ms540731.aspx
(Read 20.10.2010)
14. Microsoft Office Excel 2007 Top 10 Benefits (n.d)
http://office.microsoft.com/en-us/excel/microsoft-office-excel-2007-top-10-benefits-HA010164179.aspx
(Read 3.3.2011)
15. Microsoft SharePoint Server (27.3.2011)
http://en.wikipedia.org/wiki/Microsoft_SharePoint_Server
(Read 8.8.2010)
16. Microsoft Visual Studio 2010 programming Excel (n.d)
http://msdn.microsoft.com/en-us/library/bb386107.aspx
(Read 3.3.2011)
17. Planning libraries and lists in SharePoint 2007 (9.1.2009)
http://www.techrepublic.com/blog/datacenter/planning-libraries-and-lists-in-sharepoint-2007/488
(Read 15.1.2011)
18. SOAP (2.4.2011)
http://en.wikipedia.org/wiki/SOAP
(Read 7.7.2010)
19. TDiK Technical Design (25.5.2009)
Wärtsilä internal documentation, DocID: DBAA868951

20. Technical service position description (15.5.2009)
Wärtsilä internal directives, position: 20004777, Superintendent
21. Universal Data Connection (2010)
http://msdn.microsoft.com/en-us/library/ms772017.aspx
(Read 4.1.2011)
22. Web Service (30.3.2011)
http://en.wikipedia.org/wiki/Web_service
(Read 5.7.2010)
23. Writing Data from a DataTable to Excel (23.7.2009)
http://msmvps.com/blogs/deborahk/archive/2009/07/23/writing-data-from-a-datatable-to-excel.aspx
(Read 18.3.2011)
24. Wärtsilä Corporation (n.d)
www.wartsila.com/
(read 10.7.2010)
25. .NET (4.4.2011)
http://www.devtopics.com/what-is-net/
(Read 18.3.2011)