

*Mads Lundt, s103439*  
*Matthias Larsen, s103437*  
*Joachim Jensen, s103430*

# **Weather data visualization**

**A Graphical User Interface (GUI) for weather  
radar and wind energy data visualization and  
analysis**

Software Technology Project, June 2012

*Mads Lundt, s103439*  
*Matthias Larsen, s103437*  
*Joachim Jensen, s103430*

# **Weather data visualization**

**A Graphical User Interface (GUI) for weather  
radar and wind energy data visualization and  
analysis**

Software Technology Project, June 2012

Weather data visualization, A Graphical User Interface (GUI) for weather radar and wind energy data visualization and analysis

**This report was prepared by**

Mads Lundt, s103439

Matthias Larsen, s103437

Joachim Jensen, s103430

**Supervisors**

Pierre Pinson//Pierre-Julien Trombe

---

Release date: June, 2012

Category: 1 (public)

Edition: First

Rights: ©Mads Lundt, Matthias Larsen, Joachim Jensen, 2012

Department of Informatics and Mathematical Modelling  
Technical University of Denmark  
Asmussens Alle building 305  
DK-2800 Kgs. Lyngby  
Denmark

[www.imm.dtu.dk](http://www.imm.dtu.dk)

Tel: (+45) 45 25 33 51

Fax: (+45) 45 88 26 73

E-mail: [reception@imm.dtu.dk](mailto:reception@imm.dtu.dk)

# Acknowledgements

The solutions provided by this report to the Weather data visualization – A Graphical User Interface (GUI) for weather radar and wind energy data visualization and analysis, has been developed by Mads Lundt (*s103439*), Matthias Larsen (*s103437*) and Joachim Jensen (*s103430*).

– Some work distribution perhaps?

# Abstract, Résumé

A platform to visualize different kind of weather data

# Contents

<b>Abstract</b>	<b>ii</b>
<b>Contents</b>	<b>iii</b>
<b>List of Figures</b>	<b>v</b>
<b>List of Tables</b>	<b>vi</b>
<b>List of Source code</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Related work</b>	<b>2</b>
<b>3 Body</b>	<b>3</b>
3.1 The use of frameworks . . . . .	3
3.2 The C++ application . . . . .	3
3.2.1 Why a C++ application? . . . . .	3
3.2.2 Why Qt? . . . . .	3
3.2.3 Return codes . . . . .	3
3.2.4 Extensibility . . . . .	4
3.3 The PHP site . . . . .	4
3.3.1 Why PHP? . . . . .	4
3.3.2 Why FuelPHP? . . . . .	4
3.3.3 Speed optimizations . . . . .	4
3.3.4 The data . . . . .	4
<b>4 Conclusion</b>	<b>5</b>
4.1 What's been implemented . . . . .	5
4.2 What's been left out . . . . .	5
<b>5 Future work</b>	<b>6</b>
<b>References</b>	<b>7</b>
<b>Appendix</b>	<b>8</b>

A Appendix	8
------------	---

## List of Figures



# List of Tables

3.1	Return codes from the C++ application . . . . .	4
-----	---	---

# Listings

# 1

## Introduction

The problem with this project is analysing the different kind of data and visualize it in a smooth and easy way. There are other weather applications on the market, but we had to create a new weather application using opensource (openstreetmap, Qt etc.). We had different kind of data in different language, CSV-files, WRK-files and NC-files. We only did the implementation for CSV-files and WRK-files. We are very focused on getting high performance and getting this optimized - that was one of the biggest challenges.

This is interesting because we have a lot of data to analyse and visualize, and still want high performance and user-friendly overview. We are combining different programming language to get it optimized.

This is tough, because we had to do some changes during its construction, such as in the beginning we didn't think about database integration. Many of these changes is to satisfy our own requirements through the project.

We have a lot of data to analyse and visualize, so we have to figure out how to do this in the best possible way.

At the beginning we looked at DMI and TV2-vejret, but none of these satisfied our requirements, that we specified at the beginning. DMI had many different data, but wasn't created in a user-friendly way and TV2-vejret did only have weather forecast. There are similar weather programs out there, but none of these have either user-friendly way of huge amount data or using opensource.

# 2

## Related work

A GUI has been developed in Matlab..

A second attempt made use of Google Maps..

# 3

## Body

### 3.1 The use of frameworks

---

Using frameworks because of the decreased development time due to already developed and tested libraries for various tasks such as database accesss, file handling and file uploading.

### 3.2 The C++ application

---

#### 3.2.1 Why a C++ application?

We're using a C++ application because it is much faster to parse huge files in C++ than PHP or JavaScript. The choice of language was because we had some knowledge of both C and C++. C++ is object oriented where as C is procedural language, and seeing as we all favour object oriented programming, the choice between the two was easy.

We could have chosen other languages such as F, F#, Java, Perl or Ruby, but our limited knowledge about these made us stick with C++, which is also quite fast and portable.

#### 3.2.2 Why Qt?

We've decided to use Qt because of the support and community around it. It's very mature and most importantly, object oriented.

#### 3.2.3 Return codes

The C++ application returns the following codes to the PHP after processing data:

Code	Explanation
0	Failure
1	Success
2	Success, unkown params ignored
10	Failed, missing fileid
11	Failed, missing filename
12	Failed, missing type
20	Unknown file type
21	File not found
22	Invalid file, or wrongly formatted
30	Could not read input file
31	Could not write to input file
32	Could not read output file
33	Could not write to output file
34	Could not delete output file

**Table 3.1:** Return codes from the C++ application

#### **3.2.4 Extensibility**

The application is coded with extensibility in mind, and all parsers is therefore extending the base class *Parser*. The *Parser* class contains the basic methods of setting the filename, opening for read and write and closing the open file handle.

### **3.3 The PHP site**

---

#### **3.3.1 Why PHP?**

#### **3.3.2 Why FuelPHP?**

#### **3.3.3 Speed optimizations**

##### **3.3.3.1 Caching**

#### **3.3.4 The data**

The application support the upload of the following file types:

**csv** Observations form a wind farm.

# 4

## Conclusion

### 4.1 What's been implemented

---

We did the implementation of CSV-files and WRK-files, in zip-file and single files. We did the database integration, to make it easier and to get higher performance. We got visualizing of CSV-files and WRK-files to work, in a way we wanted to be.

### 4.2 What's been left out

---

We didn't do the implementation of NC-files, due the troubles with CSV and WRK.

# 5

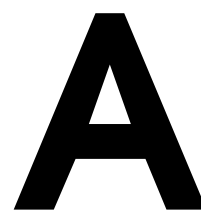
## Future work

The use of well developed and supported frameworks makes the application easy to extend to support new file formats.



## References

- [1] *Microsoft Windows Bitmap Format*. see `CampusNet/Labs/bmp format.pdf`.
- [2] *Microsoft Windows Bitmap Format*. [http://en.wikipedia.org/wiki/BMP\\_file\\_format](http://en.wikipedia.org/wiki/BMP_file_format)



## Appendix

asd



**[www.imm.dtu.dk](http://www.imm.dtu.dk)**

Department of Informatics and Mathematical Modelling  
Technical University of Denmark  
Asmussens Alle building 325  
DK-2800 Kgs. Lyngby  
Denmark  
Tel: (+45) 45 25 33 51  
Fax: (+45) 45 88 26 73  
E-mail: [reception@imm.dtu.dk](mailto:reception@imm.dtu.dk)