Technical University of Denmark

Matthias Larsen, s103437
Joachim Jensen, s103430

UML

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

02350 Windows Programming using C# and .Net, December 2012

DTU Informatics
Department of Informatics and Mathematical Modelling

Matthias Larsen, s103437
Joachim Jensen, s103430

UML

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

02350 Windows Programming using C# and .Net, December 2012

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

This report was prepared by
Matthias Larsen, s103437
Joachim Jensen, s103430

Supervisors
Bjarne Poulsen

Release date:	December, 2012
Category:	1 (public)
Edition:	First

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

www.imm.dtu.dk
Tel: (+45) 45 25 33 51
Fax: (+45) 45 88 26 73
E-mail: reception@imm.dtu.dk

Abstract

Data for weather analysis and forecasts consists of enormous amounts of data.

Comparing these huge amounts of data quickly becomes difficult. Most people interpret images easier than numbers, hence the need for some sort of visualization for the data sets.

The data come from multiple sources (e.g. on site observations from measuring stations, meteorological forecasts from Numerical Weather Prediction models) and, consequently have very diverse formats (time series, georeferenced data, gridded data).

These diverse formats raise an important issue because no common or efficient platform for visualizing and analyzing these data exists except for two earlier attempts: one in MATLAB and one with Google Maps. These earlier application were not flexible enough to be used in a bigger context.

This paper shows the development and considerations throughout the project of an application that might serve as the beginning of a new common open source platform for analyzing, visualizing and comparing weather related data.

Contents

List of Figures

List of Tables

Listings

1 Introduction

The problem with this project is analysing the different kinds of data and visualize it in a smooth and easy way. Other weather applications exists on the market, but one of the preferences to create the new weather application, is using open source (OpenStreetMap, Qt etc.).

The test data provided to this project is in different kinds of file formats, namely csv, wrk and NetCDF[1].

This is interesting because there is a lot of data to analyze and visualize. A combination of different programming languages will be used to get higher performance while maintaining functionality and user-friendliness.

Because of the many different file formats and huge amount of data, it is difficult to create an application that is flexible enough to handle all the data in an optimal way.

There are other applications out there, like DMI[2] and TV2-vejret[3], but none of these satisfied our requirements that was specified at the beginning. DMI had many different data, but wasn't created in a user-friendly way and TV2-vejret only have weather forecast. There are similar weather programs out there, but none of these have either user-friendly way of handling huge amount data or using open source.

This project is going to focus on extensibility, speed and user experience, as key components.

[1] Network Common Data Form - See [13]
[2] Danish Meteorological Institute - http://dmi.dk
[3] Denmark's nationwide commercial TV channel - http://vejret.tv2.dk

2 Analysis

A lot of considerations have gone into the development process of the application. The following chapter explains the considerations with the most impact on the application.

TÆLLER 5

2.1 Choice of diagram

2.1.0.1 Solution strategy

iterationsplan

2.2 Risks

Fremgangsmåden ved risikovurderinger er:

Identifikation af risici Vurdering af risici Handlinger mod risici Opfølgning

2.3 Comparison

It has been discussed if it should be possible to compare data from multiple wind farms at the same time, and how it should be done if that was the case.

One of the ideas was to let single click on a windmill put it in a 'selected' state and a double click on a windmill would trigger the comparison view between all the windmills in the 'selected' state.

The comparison view was also discussed to be either one chart, like the one implemented, but with all the data represented with different colors for each wind farm. Another way of viewing them was to have multiple, small, draggable windows, each one representing a wind farm. This would allow browsing the data for each wind farm independent of each other, but also make it annoying when wanting to compare at the same time and having to manually move back or forth in time on each window.

2.4 Time handling

As time goes by, one would expect the application to contain massive amounts of data. In the provided test data[1], the application contains ~140 rows of data for wrk files and ~73,000 rows of data for the csv file.

It was discussed how we should limit the view of data, since it would require quite a lot of processing power to display that much data.

A 'from' and 'to' field for selecting a date range that all data on the application would be limited to which would increase performance, reduce server load and increase overall UX.

[1] Data02122.tar.gz provided by Pierre-Julien Trombe on CampusNet

3

Design & Implementation

3.1 GUI

TÆLLER 35

3.2 Application layer

TÆLLER 20

3.3 Design patterns

2-3

TÆLLER 10

3.4 Tests

TÆLLER 10

3.5 Noget der ikke er udnervist i

TÆLLER 5

4 Conclusion

A weather visualization application that met the requirements was created successfully.

The implementation of different file types, data visualizing and database integration, to make it easier and to get higher performance, are completed with the help of tools and methods stated in this paper while keeping focus on extensibility, speed and user experience.

Due to lack of time, the support for NetCDF files were not implemented, but the flexibility of the software allows this to be done at a later date without breaking any existing functionality.

The final product ended up being a satisfactorily solution for a weather visualization application that can manage the enormous amounts of data in an intuitive way while being a viable candidate for a future common platform in the industry.

4.1 Future work

Besides fixing the currently known bugs stated in **??**, we have some ideas for the application, some of which have been described in 2 that did not make it to the existing application.

A

Use cases

We have two types of actors:

- User: Visitor.
- Admin: User with access to the administration panel.

A.1 Clicking on a radar at the map

Actor: User

Scenario:

- Checks if there is data for the date
- If images is not generated, it will generate them real time.
- Radar shows images over an interval.
- *One click for play/pause and double click for reset.*
- When there is no more data, the radar resets.

Alternative scenario:

1. No data is available for that moment and a message appears.

A.2 Clicking on a wind farm at the map

Actor: User

Scenario:

- Opens chart dialog.
- Chart data is retrieved in the background
- Data is visualized if available

A.3 Changing view in chart for wind farm

Actor: User

Scenario:

1. Changing to 2-week view.
    - Changing chart view interval to 2 weeks.
    - Load new data for the new interval.
2. Changing to weekly view.
    - Changing chart view interval to 1 week.
    - Load new data for the new interval.
3. Changing to daily view.
    - Changing chart view interval to 1 day.
    - Load new data for the new interval.

A.4 Changing data set in chart

Actor: User

Scenario:

- Selecting a new data set
- The data is retrieved from the server
- Visualize the data if available

Alternative scenario

- Unselecting a data set
- The data is removed from the chart

A.5 Using control buttons in chart for wind farm

Actor: User

Scenario:

1. Pressing fast backward.
    - Go two weeks back on 2-week view, one week back on weekly view and one day on daily view.

- Loads new data.

2. Pressing backward.
   - Go one week back on 2-week view, one day back on weekly view and one hour on daily view.
   - Loads new data.
3. Pressing play.
   - Change to daily view and go one hour forward every second.
   - Loads new data.
4. Pressing forward.
   - Go one week forward on 2-week view, one day forward on weekly view and one hour on daily view.
   - Loads new data.
5. Pressing fast forward.
   - Go two weeks forward on 2-week view, one week forward on weekly view and one day on daily view.
   - Loads new data.

A.6 Upload file

Actor: Admin

Scenario:

- Sign in with correct username and password.
- Add new file (CSV, WRK, ZIP) and enter correct timezone.
- If file is ZIP, it will decompress.
- File is uploaded.
- File is being parsed.

Alternative scenario:

1. Sign in with wrong username or password.
2. Adding file that is not correct format and get an error message.
3. Adding ZIP file, but files inside zip are not correct. They will be skipped.
4. File exists and will not be uploaded.

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