

# Bachelor Thesis

YuanDong Hu

Weather Data Website and Analysis of Temperature  
Extrapolation Algorithms

*Fakultät Technik und Informatik  
Department Informations- und  
Elektrotechnik*

*Faculty of Engineering  
and Computer Science  
Department of Information and  
Electrical Engineering*

YuanDong Hu

Weather Data Website and Analysis of Temperature  
Extrapolation Algorithms

Bachelor Thesis based on the examination and study regulations for  
the Bachelor of Engineering degree programme

Information Engineering

at the Department of Information and Electrical Engineering

of the Faculty of Engineering and Computer Science

of the University of Applied Sciences Hamburg

Supervising examiner : Prof. Dr. Hans-Juergen Hotop

Second examiner : Prof. Dr. Ulrich Sauvagerd

Day of delivery 24-02-2010

**YuanDong Hu**

## Title of the Bachelor Thesis

Weather Data Website and Analysis of Temperature Extrapolation Algorithms

### Keywords

Web server, Database, Weather Measured Value, TYPO3, Linux, JAVA, APPLET, FTP, Henderson Trend filter, Double Exponential Smoothing, JFreeChart

### Abstract

In this thesis, a web application and presentation of weather data from a local weather station database is described. The weather station website is established on the TYPO3 content management system. JAVA Applets are utilized to display the weather situation on website. Via a graphical user interface the user of the website can select an actual weather view, a day, a month or year presentation of ten weather values in graphical form.

Trend filter and Double exponential smoothing approach for weather data extrapolation prediction is the scientific research part of this project. This part emphasizes on predication of temperature in one specific day. The fundamental principle of two methods are described and demonstrated by temperature value extrapolation in different forms. The comparison of the results of the two methods is done at the end of thesis.

**YuanDong Hu**

### Thema der Bachelorarbeit

Wetterdaten-Website und Analyse von Extrapolationsalgorithmen für die Temperatur

### Stichworte

Web Server, Datenbank, Wetterdaten, TYPO3, Linux, JAVA, APPLET, FTP, Henderson Trendfilter, Double Exponential Smoothing Methode, JFreeChart

### Kurzzusammenfassung

In diesem Report wird eine Web-Anwendung für die Darstellung von Wetterstation einer lokalen Wetterstation, deren Daten in einer Datenbank gesammelt werden. Die Website der Wetterstation wird auf dem ContentManagement System TYPO3 entwickelt. Zu Darstellung der Wetterdaten werden JAVA Applets verwendet. Der Nutzer kann über eine grafische Benutzerschnittstelle die Darstellung von zehn Wettersignalen in graphischer Form als aktuelle Wetterdaten, Daten eines Tages, eines Monats oder eines Jahres auswählen.

Der wissenschaftliche Teil der Arbeit befasst sich mit der Untersuchung eines Trendenfilters und der Double exponential smoothing Methode zur Extrapolation von Wetterdaten. In diesen Teil wird die Temperaturentwicklung eines Tages untersucht. Dabei wird das Grundprinzip der beiden verwendeten Methoden beschrieben und am Beispiel der Temperaturvoraussage für einen Tag die Ergebnisse der Verfahren vorgestellt. Am Ende der Arbeit werden die Ergebnisse der beiden Methoden miteinander verglichen.

# Contents

<b>1. Introduction .....</b>	<b>7</b>
<b>2. Weather Station .....</b>	<b>8</b>
2.1 Description of Weather Station (MWS 6) .....	8
2.2 Description of Measured Weather Parameters .....	9
<b>3. Software Description.....</b>	<b>11</b>
3.1 Database.....	11
3.1.1 Characteristic of Relational Database .....	12
3.1.2 SQL Anywhere 10.....	14
3.1.3 SQL Anywhere Server Programming .....	16
3.2 TYPO3 .....	17
3.2.1 Content Management System.....	18
3.2.2 TYPO3 Core Feature .....	18
3.3 JAVA Technologies .....	21
3.3.1 Eclipse Platform Configuration .....	21
3.3.2 JAVA Applet .....	22
3.3.3 JFreeChart.....	23
<b>4. Weather Station Website Requirements .....</b>	<b>25</b>
4.1 Weather Database .....	25
4.2 JDBC programs and File System .....	26
4.3 Interactive GUI .....	28
4.4 Three-tier Architecture .....	28
4.5 JAVA Programs compile and execution on the Server.....	30
<b>5. Weather Station Website Design .....</b>	<b>31</b>
5.1 Weather Station User Case Diagram .....	32
5.2 Weather Station Class Diagram .....	34
5.3 Weather Station Homepage Design .....	36
5.4 Weather Station Database GUI Design.....	38
5.5 JDBC programs Design.....	40

<b>6. Weather Station Website Results</b> .....	<b>44</b>
<b>7. Analysis Temperature Extrapolation Algorithms</b> .....	<b>49</b>
7.1 Forecasting Methods.....	49
7.2 The Principle of Trend Filter.....	51
7.3 Henderson Trend Filter for Temperature Extrapolation.....	52
7.3.1 13-term Henderson Trend-cycle Filter.....	52
7.4 Double Exponential Smoothing for Temperature Extrapolation.....	55
7.5 Algorithms Design for Temperature Extrapolation.....	57
7.5.1 Temperature Extrapolation Class Diagram.....	58
7.5.2 Henderson 13-term Trend Filter Design.....	59
7.5.3 Double Exponential Smoothing Design.....	60
7.6 Temperature Extrapolation Results.....	61
<b>8. Major Challenges encountered</b> .....	<b>65</b>
<b>9. Conclusion</b> .....	<b>66</b>
<b>10. References</b> .....	<b>67</b>
<b>11. Appendix</b> .....	<b>69</b>

## List of Figures

1. Internal Image of MWS 6
2. MWS 6 Weather Station
3. Measuring Method for Wind Direction and Wind Speed
4. Sybase Central Console
5. Interactive SQL Console
6. schematic of connection to database
7. TYPO3 content management systems
8. page tree and content schematic
9. content element wizard
10. Sample Charts Plot by JFreeChart
11. Three-Tier Architecture
12. Three tier architecture of Weather station
13. Weather station user case diagrams
14. Class diagram of weather station
15. Sequence diagram of readSQL() method
16. Sequence diagram of readDayFile method
17. Sequence diagram of select60DaysWeatherData
18. sequence diagram of readContent method
19. Homepage of weather station
20. sitemap and forecast widget of weather station
21. Weather database GUI
22. Demo for Temperature and Barometer chart on 1-2-2010
23. Information of MWS 6 weather station
24. The weights of 13-term Henderson filter I/C ratio of 3.5
25. Class diagram of Temperature Extrapolation
26. Sequence diagram of algorithm of Henderson 13-term Trend filter
27. Sequence diagram of Double Exponential Smoothing
28. Temperature extrapolation per hour on 20-Feb-2009
29. Temperature extrapolation per hour on 20-July-2009
30. Temperature extrapolation per 20 minutes on 20-Feb-2009
31. Temperature extrapolation per 20 minutes on 20-July-2009

## 1. Introduction

Weather is a set of all the phenomena occurring in a given atmosphere at a given time. Most weather phenomena occur in the troposphere, just below the stratosphere. Weather refers, generally atmospheric day-to-day temperature and precipitation activity, whereas climate is the term for the average atmospheric conditions over longer periods of time.

Weather takes a great role to people on earth. It is particularly useful to the experts working in the weather forecasting field such as Airline industry, farmers, and disaster response teams. It is extremely important for the weather data to be recorded accurately and interpret in a human readable manner.

Weather forecasting system is the application of science and technology to predict the state of the atmosphere for a future time and a given location. Weather forecasts are made by collecting quantitative data about the current state of the atmosphere and using scientific understanding of atmospheric processes to project how the atmosphere will evolve. Forecast models which include data collection, data assimilation and analysis, Numerical weather prediction and Model output post processing before being presented as the forecast are now used to determine the future conditions. Completeness, accuracy, instantaneity are three vital factors for accurate weather forecasting.

This thesis seeks to discuss how weather data that is exactly recorded and stored in a database is presented on web pages in an interactive way to the user. The whole web site is established on TYPO3 Content Management Framework which consists of a central program core with only limited functionality. To extend this core system there are hundreds of extensions which in combination will provide a very capable system. JAVA applets which are used to provide interactive features to web applications are utilized and embed into the html pages that are constructed by TYPO3.

The second part of this thesis endeavor upon finding out algorithms that approximately predicate temperatures in one specific day. The original temperatures and estimated temperature will be plotted in one xyplot graphic. The comparison between original temperatures and estimated temperatures is done by computing the sigma of mean value of temperatures.

Two Algorithms are introduced and explained in the thesis. The Henderson Trend filter and double exponential smoothing are two of the trial algorithms used to extrapolate temperature in one day according to previous temperatures. The comparisons between two methods will be done at the end of thesis and which method makes more sense to the temperatures in one day has to be determined.

## 2. Weather Station

A weather station is a facility with instruments and equipment for observing atmospheric conditions to provide information for weather forecasts and to study the weather and climate. The measurements taken include temperature, barometer, humidity, wind speed, wind direction and precipitation amounts. Typical weather stations have several instruments including Thermometer, Barometer, Hygrometer, Anemometer, Wind vane, Rain gauge.

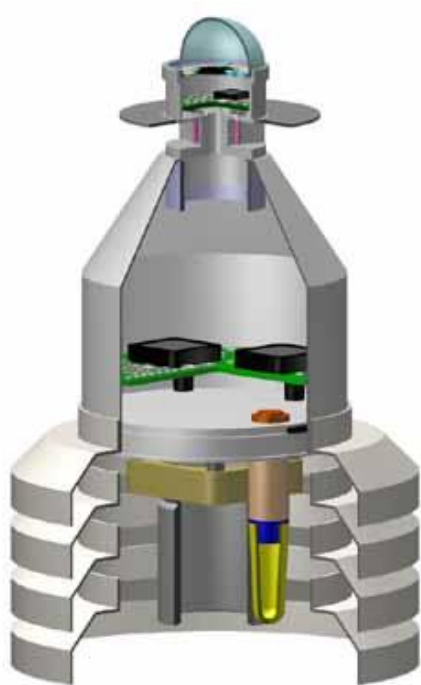


Figure 1 Internal Image of MWS 6



Figure2 MWS 6 Weather station

### 2.1 Description of the Weather station MWS 6

The weather station located on the roof of university is MWS 6 come from Reinhardt – Testsysteme. The internal design is figure 1 and the outlook of MWS 6 is figure 2 and. MWS 6 is a Microprocessor –Weather Station with 6 Sensors for 6 Measuring parameters as following:

- Temperature
- Atmospheric pressure
- Humidity

° Celsius  
hectoPascal  
%



- Wind Speed km/h
- Wind Direction °
- Solar energy Watt/m<sup>2</sup>

The MWS 6 weather station has a special method of measuring wind direction and wind speed (0 to 50km/h): Three wind speed sensors record the necessary values.

Wind maximum peak, Wind average peak, wind chill factor and prevalent wind direction are measured. A pyranometer for measuring global radiation (solar energy) is standard as temperature, relative humidity, dew point and barometric pressure. Data are transmitted via RS232 –interface. With a TCP/IP module data transmission is available via an existing network.

## 2.2 Description of measured weather parameters

**Temperature** is the average energy in each degree of freedom of the particles in a thermodynamic system. It is a physical property that underlies the common notions of hot and cold. Something that feels hotter generally has a higher temperature, though temperature is not a direct measurement of heat. Temperature is measured with thermometers that may be calibrated to a variety of temperature scales. Celsius scale is used for most temperature measuring purposes. In MWS 6, temperature measurement is based on a temperature sensor and linearised by microprocessor. Its range is from -40°C to +50°C. Measuring accuracy is ±0.3°C.

**Atmospheric pressure** is defined as the force per unit area exerted against a surface by the weight of air above that surface at any given point in the Earth's atmosphere. Low pressure areas have less atmospheric mass above their location, whereas high pressure areas have more atmospheric mass above their location. Atmospheric pressure is measured by Barometer which is done by the pressure sensor in MWS6. Pressure sensor can be used in the temperature range of -40°C to +50°C. Absolute measuring range is from 700hPa to 1100hPa with ± 1hPa accuracy.

**Wind Direction** is the direction from which a wind originates. It is usually reported in cardinal directions or in azimuth degree. Modern instruments used to measure wind speed and direction are called anemometers and wind vanes respectively. In MWS 6, wind direction is measured by wind direction sensor. It consists of three glass-coated PT100 shifted in an angle of 120°, which are heated to 50-60°C and cooled by wind. A microprocessor calculates the wind direction with 90° being East, 180°being South,270°being West and 0° being North. Measuring accuracy is 5° with a starting speed by 0.6m/s.

**Wind Speed** is the speed of wind, the movement of air or other gases in an atmosphere. Wind speed has always meant the movement of air in an outside environment. It is

measured with wind speed sensor. The range of wind speed is 0 km/h to 150 km/h with a measuring accuracy of  $\pm 2$  km/h, starting speed is 0.6 m/s. In MWS 6 there is 3-fold way of measuring wind speed with current wind speed (WG), average wind speed (WD) and wind peaks (WS).

### MWS 6 Method for Measuring Wind

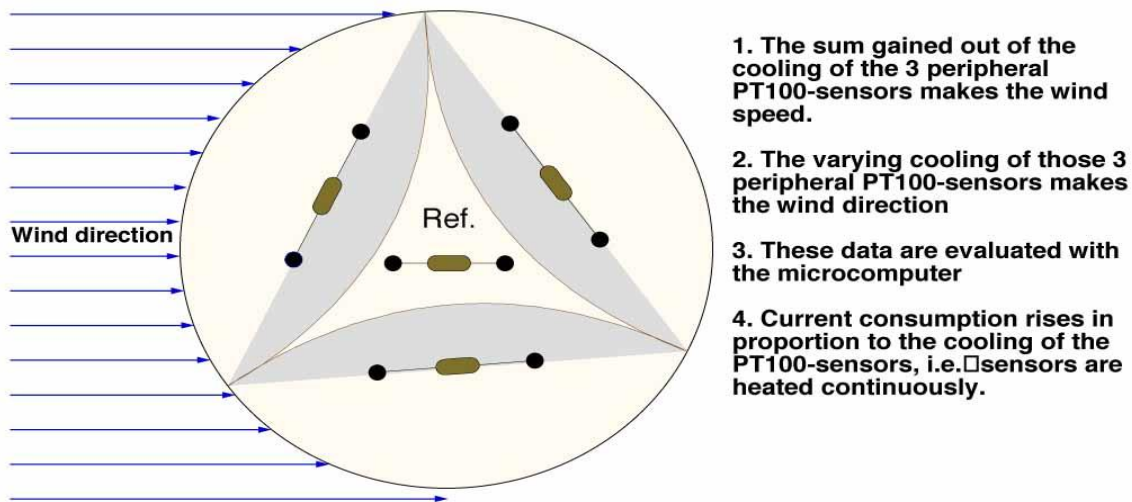


Figure 3 Measuring Method for Wind Direction and Wind Speed

**Maximum Peak Wind Speed** is the maximum instantaneous wind speed measured since the last routine observation.

**Average Peak Wind Speed** is the mean of all the maximum peak wind speeds observed over given fixed intervals for a given time.

**Wind chill Factor** is the apparent temperature felt on exposed skin due to wind. The degree of this phenomenon depends on both air temperature and wind speed. The wind chill temperature (wind chill factor) is always lower than the air temperature for values where the wind chill formula is valid.

**Prevalent Wind Direction** is the direction from which wind blows most frequently across a certain region within a given time period. Different regions on Earth have different prevalent wind directions, which are dependent on the nature of the general circulation of the atmosphere and the latitudinal wind zones. The periods most frequently used are the observational day, month, season, and year. Methods of determination vary from a simple count of periodic observations to the computation of a wind rose.

**Solar radiation** is the visible and near-visible (ultraviolet and near-infrared) radiation emitted from the sun. It is measured by Solar Sensor in MWS 6 (pyranometer) which

absorbs radiation between 305 and 2600 nm. Its range is from 0 to 1500 W/m<sup>2</sup>. Measuring accuracy is  $\pm 2\%$  of final value.

**Humidity** is the amount of water vapor in the air. Relative humidity is an important metric used in forecasting weather. Relative humidity is defined as the ratio of the partial pressure of water vapor in a parcel of air to the saturated vapor pressure of water vapor at a prescribed temperature.

The device used to measure humidity is called a Humidity Sensor (hygrometer). The humidity sensor can be used in a temperature range between -40°C to +50° C. It is linearised to an accuracy of 2%. Its range is from 12 to 100%. Measuring accuracy is  $\pm 2\%$  and display also as dew point measurement in °C or °F.

Above ten weather parameters are stored in the Sybase relational database which is called Weather in 60 seconds.

All the data analysis and fetching are based on this “Weather” database and ten weather parameters.

### 3. Software Description

In order to present the weather data on web pages, several technologies should be introduced and utilized in the weather station project.

First of all, the weather data is stored in the SQL Anywhere relational database. As long as the weather data has to be plotted in a graphic or present in table form, database must be connected and DBMS (Database Management System) must be used to retrieve the data. Considering that weather data will be presented in the form of JAVA Applet, JDBC should be the bridge between Applet and relational database.

Secondly, TYPO3 is the main system where web pages should be contained inside. TYPO3 is content management system for enterprise purposes on the web and in intranets. There are some restrictions on the web pages if Applets are located on the web sites. It is simple and convenient to create a large website by TYPO3, but it needs to be extended if you want to create some stuff dedicated to own website.

The third part is web server. A web server is a computer program that delivers serves content, such as web pages using Hypertext transfer protocol. The primary function of a web server is to deliver web pages (HTML documents) and associated content (e.g. images, style sheets, JavaScript, java applet) to clients. A client, commonly a web browser, makes a request for a specific resource using HTTP. In our case, the web server is Apache HTTP Server with Linux operating system. During the implementation of project,

the java file needs to be updated to a specific directory of Apache web server and compiled on the web server via Putty.

The last part is about other technologies which also might be a good choice to implement weather station web pages such as a specific TYPO3 extension, PHP, JAVA Servlet. The explanation about the pros and cons of other technologies will be done in the last part.

## 3.1 Database

A database is an integrated collection of logically related records or files consolidated into a common pool that provides data for one or more multiple uses. A database management system (DBMS) consists of software that organizes the storage of data. A DBMS controls the creation, maintenance, and use of the database storage structures of social organization and of their users. Database management systems are usually categorized according to the database model that they support such as network, relational or object model. The model tends to determine the query languages that are available to access the database. One commonly used query language for the relational database is SQL, although SQL syntax and function can vary from one DBMS to another.

### 3.1.1 characteristics of relational database

A relational database matches data by using common characteristics found within the data set. The resulting groups of data are organized and are much easier for people to understand.

The software used to do this grouping is called a relational database management system.

Strictly, a relational database is a collection of relations (tables). Relational database theory used a set of mathematical terms, which are roughly equivalent to SQL database terminology.

A relation is defined as a set of tuples (rows) that have the same attributes (columns). A tuple often represents an object and information about that object. Objects are typically physical objects or concepts. A relation is usually described as a table, which is organized into rows and columns. All the data referenced by an attribute are in the same domain and conform to the same constraints. Application access data by specifying queries, which use operations such as **select** to identify tuples, **project** to identify attributes, and **join** to combine relations. Relations can be modified using the **insert**, **delete**, and **update** operators. New tuples can supply explicit values or be derived from a query. Si-

Similarly, queries identify tuples for updating or deleting. It is necessary for each tuple of a relation to be uniquely identifiable by some combination (one or more) of its attribute values. This combination is referred to as the **primary key**.

In a relational database, all data are stored and accessed via relations. Relations that store data are called “base relations”, and in implementation are called “tables”. Other relations do not store data, but are computed by applying relational operations to other relations. These relations are sometimes called “derived relations”. In implementation these are called “views” or “queries”. Derived relations are convenient in that though they may grab information from several relations, they act as a single relation. Also, derived relations can be used as an abstraction layer.

**Domain** describes the set of possible values for a given attribute. Because a domain constrains the attribute’s values and name, it can be considered a constraint. Mathematically, attaching a domain to an attribute means that “all values for this attribute must be an element of the specified set”.

**Constraints** allow you to further restrict the domain of an attribute. They restrict the data that can be stored in relations. These are usually defined using expressions that result in a Boolean value, indicating whether or not the data satisfies the constraint. Constraints can apply to single attributes, to a tuple or to an entire relation.

A **primary key** uniquely defines a relationship within a database. In order for an attribute to be a good primary key it must not be repeated. While natural attributes are sometimes good primary keys. Surrogate keys are often used instead. A surrogate key has no intrinsic meaning, but rather is useful through its ability to uniquely identify a tuple.

Another common occurrence, especially in regards to N: M cardinality is the composite key. A composite key is a key made up of two or more attributes within a table that uniquely identify a record.

A **foreign key** is a reference to a key in another relation, meaning that the referencing tuple has, as one of its attributes, the values of a key in the referenced tuple. Foreign keys need not have unique values in the referencing relation. They effectively use the values of attributes in the referenced relation to restrict the domain of one or more attributes in the referencing relation.

A **stored procedure** is executable code that is associated with, and generally stored in, the database. Stored procedures usually collect and customize common operations, like inserting a tuple into a relation, gathering statistical information about usage patterns, or encapsulating complex business logic and calculations.

An index is one way of providing quicker access to data. Indices can be created on any combination of attributes on a relation. Queries that filter using those attributes can find

matching tuples randomly using the index, without having to check each tuple in turn. Relational databases typically supply multiple indexing techniques, each of which is optimal for some combination of data distribution, relation size, and typical access pattern.

### 3.1.2 SQL Anywhere 10

**SQL Anywhere** is a Relational Database management System product from **iAnywhere** Solutions. **iAnywhere** is a subsidiary of **Sybase**. SQL Anywhere is a comprehensive package that provides technologies for data management and enterprise data exchange, enabling the rapid development of database-powered applications for server, desktop, mobile, and remote office environments.

**SQL Anywhere** offers:

- Data management technologies
- Data exchange technologies
- Design and management tools

**Relational database system:**

SQL Anywhere offers two databases: SQL Anywhere Server and UltraLite.

SQL Anywhere Server provides enterprise-caliber functionality including full transaction processing, referential integrity, materialized views, snapshot isolation, high availability via database mirroring and server clustering, SQL and JAVA stored procedures, triggers, row-level locking, automatic event scheduling, automatic backup and recovery, and much more.

For environments that demand smaller data-driven applications. The UltraLite database is ideal. UltraLite is a full relational database management system designed specifically to minimize memory and system requirements for deployment to handhelds and other mobile devices.

**Data exchange technologies:**

SQL Anywhere offers a wide range of options for exchanging data with enterprise systems and mobile devices, including such tools as:

**MobiLink –synchronization** offers session-based, bi-directional synchronization. It is ideal for exchanging data between a central database and many remote UltraLite or

SQL Anywhere databases, or between a central, non-relational data source and many remote UltraLite or SQL Anywhere databases.

## Design and Management tools:

**Sybase Central – centralized control and administration** (Figure.4) is an integrated database administration and development tool that provides access to database settings, properties, and utilities in a graphical user interface. Via plug-ins, Sybase Central can be used to manage SQL Anywhere Server, UltraLite, MobiLink, QAnywhere and the other Sybase products.

In addition to facilitating routine tasks, Sybase Central also provides performance statistics, procedure profiling, and stored procedure debugging, as well as the management of events and schedules, web services and connection profiles.

Sybase central helps administer any tasks that are performed by send SQL statements to the database server, or performed by SQL Anywhere utilities. A variety of Sybase Central tools are available to help you analyze and monitor the current performance of your SQL Anywhere database. These tools include procedure profiling, graphical plans, query execution, the Performance Monitor, request logging and time utilities. In addition, Sybase Central offers:

- Application profiling using the Application Profiling wizard
- Advanced application profiling in Application Profiling mode
- Interactive SQL – SQL query editor (Figure 5)
- MobiLink Monitor – synchronization monitoring
- Database utilities
- InfoMaker
- PowerDesigner Physical Data Model
- Data Window.Net

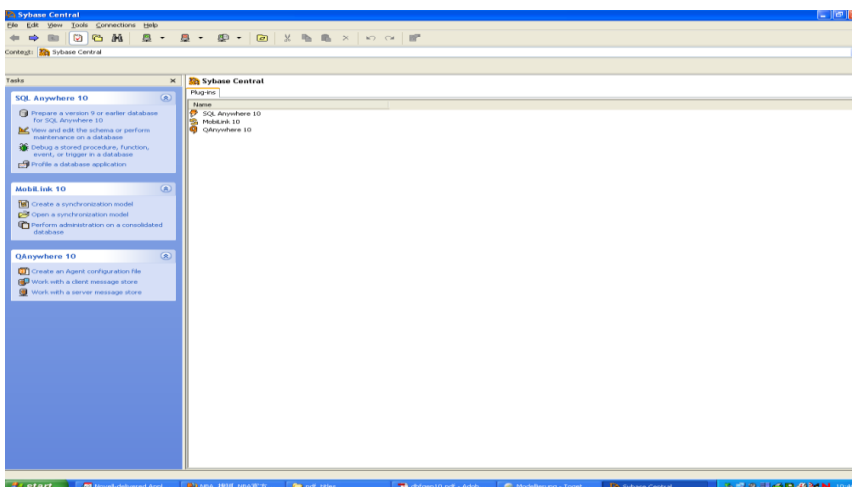


Figure 4 Sybase Central

The screenshot shows the Sybase Central SQL console window titled 'Weather (weatherread) on PCPool'. The SQL Statements pane contains the following query:

```
select fIdTemperature, fIdBarometer, fIdWindDirection, fIdSolarRadiation, fIdHygrometer, fIdWindSpeed, fIdWindMaximumPeak, fIdWindAveragePeak, fIdWindChillFactor, fIdPrevailW
```

The Results pane displays a table with 33 rows and 10 columns. The columns are: fIdTemperature, fIdBarometer, fIdWindDirection, fIdSolarRadiation, fIdHygrometer, fIdWindSpeed, fIdWindMaximumPeak, fIdWindAveragePeak, fIdWindChillFactor, and fIdPrevailW. The data shows various weather readings over time.

	fIdTemperature	fIdBarometer	fIdWindDirection	fIdSolarRadiation	fIdHygrometer	fIdWindSpeed	fIdWindMaximumPeak	fIdWindAveragePeak	fIdWindChillFactor	fIdPrevailW
1	-2.07	994.00	61.42	0.00	82.48	10.92	17.59	11.60	-7.08	
2	-2.09	994.00	63.29	0.00	82.25	11.26	17.59	11.60	-7.10	
3	-2.08	994.00	68.69	0.00	82.12	7.94	17.59	11.57	-7.04	
4	-2.08	994.00	65.46	0.00	81.94	11.78	17.59	11.50	-7.00	
5	-2.09	994.00	67.26	0.00	81.90	9.82	17.59	11.50	-7.00	
6	-2.09	994.00	61.36	0.00	81.63	11.86	17.59	11.50	-7.00	
7	-2.09	994.00	68.34	0.00	81.74	9.61	17.59	11.50	-7.00	
8	-2.09	994.00	63.12	0.00	81.76	12.38	17.59	11.50	-6.99	
9	-2.09	994.00	70.32	0.00	81.68	9.16	17.59	11.50	-6.99	
10	-2.03	994.00	63.63	0.00	81.72	10.14	17.59	11.40	-6.90	
11	-2.09	994.00	63.91	0.00	81.49	12.28	17.59	11.39	-6.94	
12	-2.09	994.00	63.17	0.00	81.40	12.14	17.59	11.39	-6.98	
13	-2.09	994.00	65.47	0.00	81.28	10.88	17.59	11.39	-6.97	
14	-2.09	994.00	64.65	0.00	81.57	11.40	14.36	10.89	-6.52	
15	-2.06	994.00	58.77	0.00	81.05	8.00	14.09	9.90	-5.62	
16	-2.00	994.00	67.35	0.00	81.20	9.58	14.15	9.28	-5.06	
17	-2.00	994.00	65.00	0.00	81.29	11.02	14.44	9.51	-5.28	
18	-2.09	994.00	63.12	0.00	80.97	12.48	15.59	10.21	-5.92	
19	-2.09	994.00	63.60	0.00	80.98	12.96	15.70	10.61	-6.29	
20	-2.09	994.00	63.55	0.00	80.93	11.02	15.70	10.96	-6.57	
21	-2.09	994.00	65.19	0.00	81.15	9.94	15.70	10.66	-6.33	
22	-2.09	993.99	59.98	0.00	80.97	12.41	15.70	10.76	-6.38	
23	-2.09	994.00	63.08	0.00	81.16	12.00	15.70	10.88	-6.51	
24	-2.09	993.90	61.69	0.00	80.92	13.67	16.43	11.11	-6.72	
25	-2.09	994.00	57.90	0.00	80.86	13.20	16.90	11.34	-6.90	
26	-2.09	994.00	57.75	0.00	81.19	10.97	16.90	11.39	-6.97	
27	-2.09	994.00	63.00	0.00	81.27	11.02	16.90	11.32	-6.89	
28	-2.09	994.00	59.11	0.00	80.94	13.38	16.90	11.40	-6.96	
29	-2.09	994.00	61.26	0.00	81.00	12.84	16.90	11.50	-7.08	
30	-2.09	994.00	54.68	0.00	81.18	10.70	16.90	11.50	-7.05	
31	-2.09	994.00	61.14	0.00	81.23	11.77	16.90	11.48	-7.00	
32	-2.09	993.97	62.07	0.00	81.19	10.12	16.90	11.47	-7.03	
33	-2.09	994.00	59.12	0.00	81.08	11.39	16.90	11.39	-6.99	

Figure 5 Interactive SQL console

### 3.1.3 SQL Anywhere Server – Programming

**JDBC** is a call-level interface for java applications. Developed by Sun Microsystems, JDBC provides java programmers with a uniform interface to a wide range of relational databases, and provides a common base on which higher level tools and interfaces can be built.

SQL Anywhere includes a pure java JDBC driver, named Sybase jConnect. It also includes the iAnywhere JDBC driver, which is a type 2 driver.

In addition to using JDBC as a client side application programming interface, JDBC can also be used inside the database server to access data from java in the database.

Java applications can be developed by using the JDBC API to connect to SQL Anywhere. Several of the applications supplied with SQL Anywhere use JDBC, such as debugger, Sybase Central, and Interactive SQL.

Figure 6 is the schematic of connection to SQL Anywhere database. Java programs establish the connection between tables in database and themselves by JDBC.



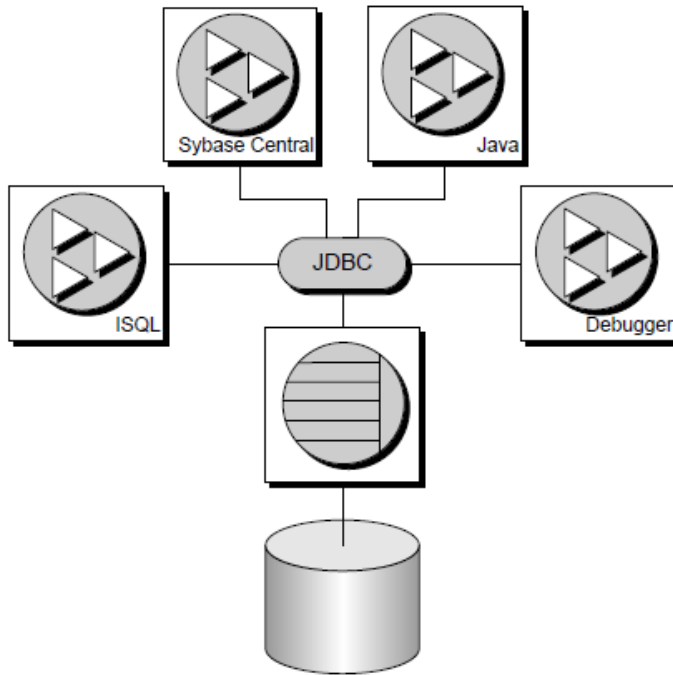
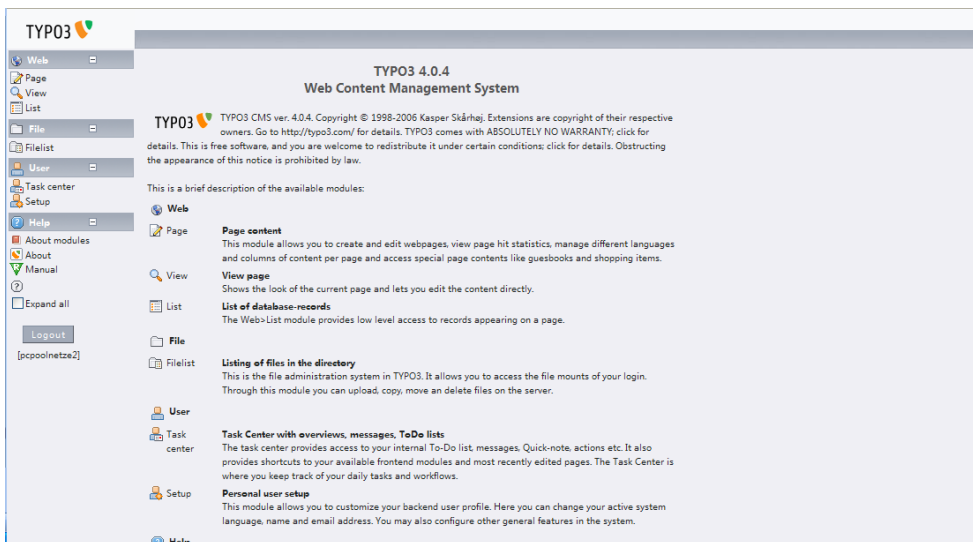


Figure 6 schematic of connection to database

### 3.2 TYPO3

Typo3 is a free and open source content management system as well as a Model-view-controller (MVC) Web Application Development framework written in PHP. Typo3 CMS is already installed on the HAW web server. Figure 5 is the overview of TYPO3



*Figure 7 TYPO3 content management systems*

TYPO3 is released under the GNU general public license. It can run on Apache or IIS on top of Linux, Microsoft windows, OS/2 or Mac OS X.

### **3.2.1 Content Management System**

A Content Management System is a collection of procedures used to manage work flow in a collaborative environment. These procedures can be manual or computer-based. The procedures are designed to:

- Allow for a large number of people to contribute to and share stored data
- Control access to data, based on user roles. User roles define what information each user can view or edit.
- Aid in easy storage and retrieval of data
- Reduce repetitive duplicate input
- Improve the ease of report writing
- Improve communication between users

In a CMS, data can be defined as almost anything – documents, movies, pictures, CMSs are frequently used for storing, controlling, revising, semantically enriching, and publishing documentation. There are six main categories of CMS, with their respective domains of use:

- Enterprise CMS
- Web CMS
- Document Management System
- Mobile Content Management System
- Component Content Management System
- Media Content Management System

### **3.2.2 TYPO3 Core Features**

TYPO3 is a sort of web content management system software, implemented as a Web application, for creating and managing HTML content. It is used to manage and control a large, dynamic collection of Web material (HTML documents and their associated images).

The software provides authoring tools designed to allow users with little knowledge of programming languages or markup languages to create and manage content with relative ease.

Most systems use a database to store content, metadata, or artifacts that might be needed by the system. Content is frequently, but not universally, stored as XML, to facilitate reuse and enable flexible presentation options.

Along with a set of ready-made interfaces, functions and modules TYPO3 has a large repository of extensions, allowing flexibility and extendibility. TYPO3 has a web frontend, which presents a TYPO3 based website to its users, along with a web based backend, used by authors and site administrators to manage content for the website.

TYPO3 can run on Apache or IIS on top of Linux, Microsoft windows or Mac OS X. It uses PHP 4 or 5 (TYPO3 4.2.0 all releases will require at least PHP 5.2) and Any database system supported by the TYPO3 DBAL including MySQL, Oracle, PostgreSQL and so on. The system can be run on any web server, with a modern CPU and 256 MB RAM. The frontend can be displayed in browsers such as Mozilla Firefox on any OS with JAVASCRIPT.

The system is based on **templates**. People can choose an existing template and change features such as logo, color and fonts, or they can construct their own templates using configuration language called **TypoScript**. In this simple notation, a large object tree structure of information is created with placeholders for data from the database to fill in.

Preprogrammed objects are configured by changing or adding values or features. The top level object is the PAGE object. There are various types of **MENU** objects.

The content is mainly stored in two table called **pages** and another called **tt\_content** which holds the elements contained in a page. Every page object has a unique identification key (uid) and is linked to a parent page. Thus, the pages are organized in tree form, allowing the system to easily generate menus and site maps. Figure 8 on the left side is the page trees hierarchy and right side is page content including page properties, layout of page and language options. One key feature of TYPO3 is that it has a flexible application programming interface which enables independent developers to provide extra functionality.

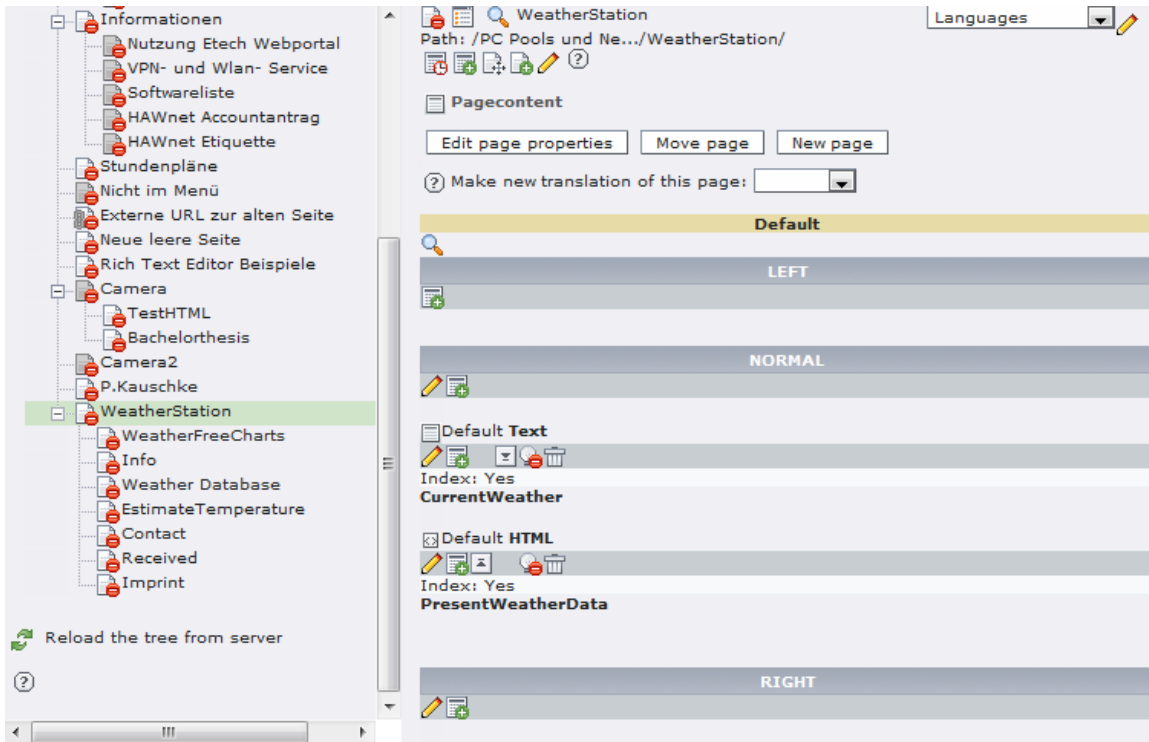


Figure 8 page tree and content schematic

The layout of web page can be modified by the existing **Templates** in TYPO3 Figure 9 is the selector box is the property of web page ,which allows users to manipulate the layout of web pages.

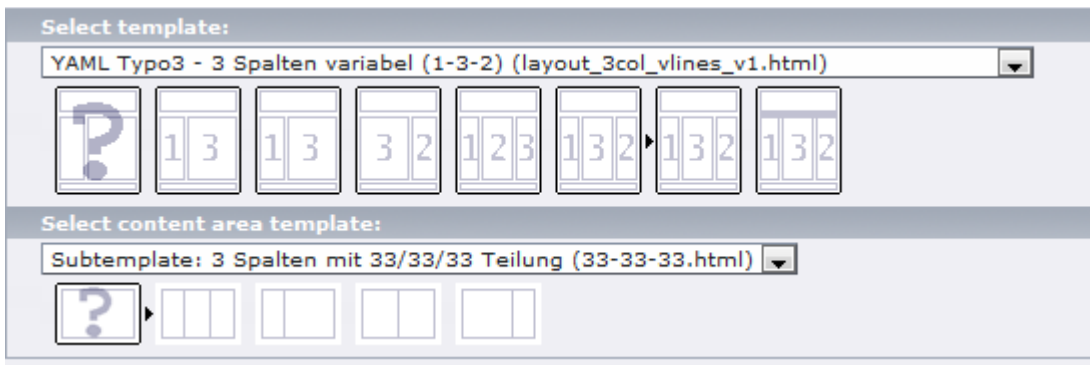


Figure 9 Template resources of TYPO3

In TYPO3 content elements are made up of web pages after the layout of web pages is decided. Figure 10 shows the exiting content elements resources of TYPO3 such as typical page content, special elements, form elements, plug-in elements. These elements can be inserted, modified, constructed in any location of a web page as long as the layout of web page is chosen.

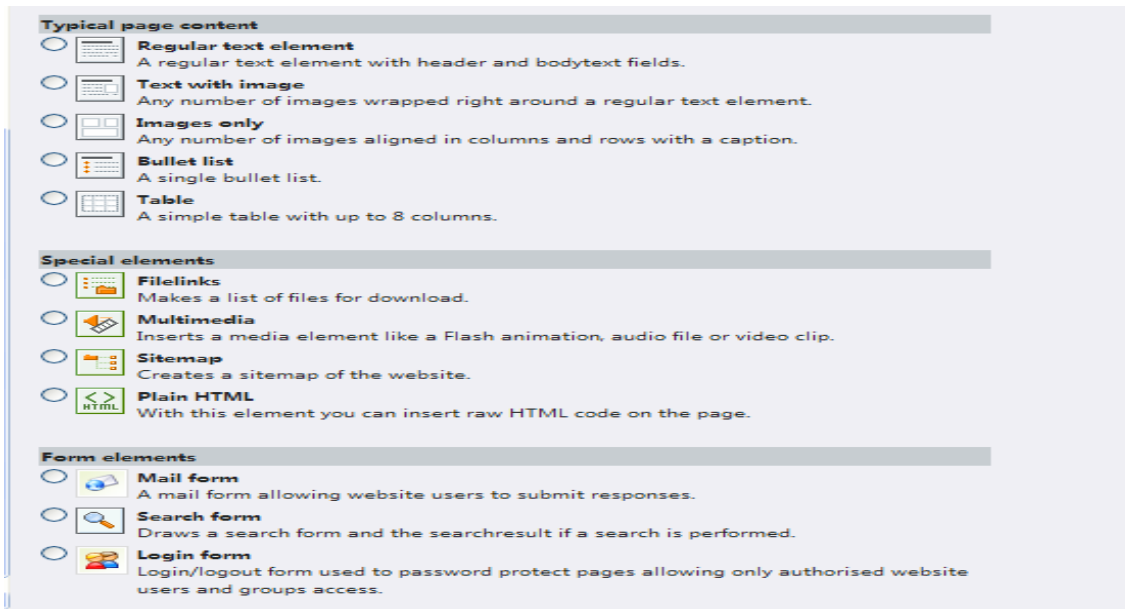


Figure 10 content element wizard

### 3.3 JAVA Technologies

Typo3 provides a convenient and interactive mean to construct a website. Static text page with image and video, sitemap, search form, mail form and news box. All the elements could be easily created, inserted and edited by the backend user, but the disadvantages is that this kind of website lack of interaction with web browsers. Java applet from java web technology will be an appropriate choice to construct a dynamic web site on top of TYPO3. Java applets are programmed by java language on the Eclipse Platform.

#### 3.3.1 Eclipse Platform configuration

Eclipse is a multi-language software development environment comprising an IDE and a plug-in system to extend it. It is written primarily in JAVA and can be used to develop applications in JAVA can be used to develop applications in java and by means of the various plug-ins, in other languages as well.

Eclipse employs plug-ins in order to provide all of its functionality on top of the runtime system. The plug-in mechanism is a lightweight software component framework. In addition to allow Eclipse to be extended using other programming languages such as C and python, the plug-in framework allows Eclipse to work with typesetting languages like LaTeX, networking application such as telnet, and database management systems.

The Eclipse SDK includes the Eclipse Java Development Tools (JDT), offering an IDE with a built-in incremental Java compiler and a full model of the java source files. The IDE also makes use of a workspace, in this case a set of metadata over a flat file space allowing external file modifications as long as the corresponding workspace “resource” is refreshed afterward. The Visual Editor project allows interfaces to be created interactively, thus allowing eclipse to be used as a RAD tool.

Eclipse’s widgets are implemented by a widget toolkit for java called SWT, unlike most java applications, which use the java standard Abstract Window Toolkit (AWT) or Swing.

Eclipse’s user interface also uses an intermediate GUI layer called JFace, which simplifies the construction of applications based on standard Widget Toolkit which is a graphical widget toolkit for use with the java platform. It is an alternative to the AWT and Swing java GUI toolkit provided by Sun Microsystems as part of the java Platform, Standard Edition.

### **3.3.2 JAVA APPLET**

A java applet is an applet delivered to the users in the form of java byte code. Java applets can run in a Web browser using a Java Virtual Machine (JVM), or in Sun’s Applet Viewer, a stand-alone tool for testing applets. Java applets are usually written in the java programming language.

Applets are used to provide interactive features to web application that cannot be provided by HTML alone. They can capture mouse input and also have controls like buttons or check boxes. In response to the user action an applet can change the provided graphic content. This makes applets well suitable for demonstration, visualization and teaching.

An applet can be a text area only, providing, for instance, a cross platform command-line interface to some remote system. If needed, an applet can leave the dedicated area and run as a separate window. However, applets have very little control over web page content outside the applet dedicated area, so they might be less useful for improving the site appearance in general.

Java applets run at a speed that is comparable to (but generally slower than) other compiled languages such C++, but many times faster than JavaScript. In addition they can use 3D hardware acceleration that is available from java. This

Makes applets well suited for non trivial, computation intensive visualization.

HTML page may embed parameters that are passed to the applet. Hence the same applet may appear differently depending on that parameters were passed. The first implementation involved downloading an applet class by class. While classes are small files,

there are frequently a lot of them, so applets got a reputation as slow loading components. However, since *jars* were introduced, an Applet is usually delivered as a single file that has a big size (hundreds of kb to several MB).

Java applet can be executed by browsers for many platforms, including Windows, Linux, Mac OS and UNIX. It is also trivial to run java applet as an application with very little extra code. This has the advantage of running a java applet in offline mode without the need for internet browser software and also directly from the development IDE such as Eclipse, Netbean.

Java applets are executed in a sandbox by most web browsers, preventing them from access local data like clipboard or file system. The sandbox is a security mechanism for separating running programs. It is often used to execute untested code, or untrusted program from unverified third-parties, suppliers and untrusted users. The code of the java applet is downloaded from a web server and the browser either embeds the applet into a web page or opens a new window showing the applet's UI. The applet can be displayed on the web page by making use of the applet tag HTML element, or object element. This specifies the applet's source and the applet's location statistics.

A java applet has some pros and cons as following:

**Pros:**

- Simple, cross platform and most web browsers supported.
- Work on all installed version of java JDK
- Cached by most web browsers so that quickly loaded
- Can be moved from server to client
- Support the changing user state
- Developers can develop and debug an applet directly by creating main routine
- An untrusted applet has no access to the local machine and can only access the server it came from.

**Cons:**

- Java plug-in requires
- Security policy makes difficult for untrusted applet to do any client side scripting.

### 3.3.3 JFreeChart

JFreeChart is a free chart library for Java (tm) platform. It is designed for use in applications, applets, Servlet and JSP. It makes easy for developers to display professional quality charts in their application. JFreeChart extension feature set includes:

- A consistent and well-documented API, supporting a wide range of chart types;
- A flexible design that is easy to extend, and targets both server-side and client-side applications;
- Support for many output types, including Swing components, image files (including PNG and JPEG), and vector graphics file formats(including PDF, EPS and SVG);
- JFreeChart is “Open source “or, more specifically, free software. It is distributed under the terms of the GNU lesser General public license (LGPL), which permits use in proprietary applications.

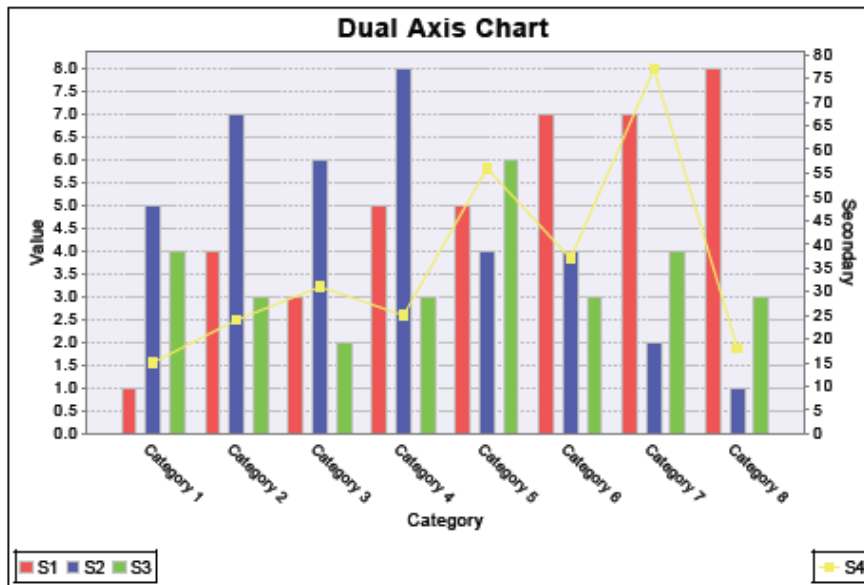


Figure.11 sample charts plot by JFreeChart

JFreeChart can generate pie charts, bar charts, line charts, scatter plots, time series charts, Gantt charts, meter charts, symbol charts, wind plots, combination charts and more.

JFreeChart has a lot of additional functionalities.

- Export charts to PNG and JPEG image file formats.
- Tooltip
- Interactive zooming
- Chart mouse events
- Annotations
- HTML image map generation
- Works in application, Servlet, JSP and applets



## 4. Weather Station Website Requirements

The requirements of weather station website consist of five components.

- Weather Database where the data are stored
- JDBC programs and file system
- Interactive weather applets
- Three-tier Architecture
- Server programming

Three-tier architecture is essential to the weather station website because it constructs the basic website architecture and the other three components located in different tier in this architecture.

### 4.1 Weather Database

The weather station is based on the roof of the building Berliner Tor 5 at the Hamburg University of applied sciences.

The weather station captures and records ten weather parameters since March 30, 2006, 5:07 pm within time intervals of 10 seconds and 60 seconds cycle mean is computed and stored in a Sybase database which is called Weather. There are 3 tables existed in the weather database:

- tblCloudInfo
- tblRawData
- tblSystem

“tblRawData” is the table, which store the weather measured value from weather station. There are 15 different fields in this table. Four fields have no specified physical meaning, so only 11 fields are required to be retrieved. All the measured value starts with “fld”.

PKey	Name	ID	DataType	Size	Scale	Nulls
√	fldRecordedAt	1	Timestamp			
	fldTemperature	2	Decimal	5	2	√
	fldBarometer	3	Decimal	7	2	√
	fldWindDirection	4	Decimal	6	2	√
	fldSolarRadiation	5	Decimal	7	2	√
	fldHygrometer	6	Decimal	6	2	√
	fldWindSpeed	7	Decimal	5	2	√
	fldWindMaximumPeak	8	Decimal	5	2	√
	fldWindAveragePeak	9	Decimal	5	2	√
	fldWindchillFactor	10	Decimal	5	2	√
	fldPrevalentWindDirection	11	Decimal	6	2	√

Table 1 DBA.tblRawData

The column fldRecordedAt is the primary key and it is protected by the trigger “PreventOKUpdateOntblRawData” to make sure the PK value cannot be modified afterwards. All the physical weather measured value has a limitation, which means all ten different fields have check constraints in the table.

- Temperature (-40 to 50 °C)
- Barometer (900.0 to 1050.0 hPa)
- WindDirection (0 to 360 °)
- SolarRadiation(0 to 1500 W/m<sup>2</sup>)
- Hygrometer(0 to 100.0 %)
- WindSpeed(0 to 60.0 km/h)
- WindMaximumPeak (0 to 60.0 km/h)
- WindAveragePeak(0 to 60.0 km/h)
- Windchillfactor(-90.0 to 90.0 °)
- PrevalentWindDirection(0 to 360.0 °)

It makes sense that all measured values have a reasonable limitation so that wrong values won't be post on the website.

Once the measured values are in the weather database, the question is how to retrieve these weather data. One possibility is via JDBC programs and file system.

## 4.2 JDBC programs and File System

JDBC is API for the java programming language that defines how a client may access a database. It provides methods for querying and updating data in a database. JDBC is oriented towards relational databases. SQL Anywhere supports the two JDBC drivers: iAnywhere JDBC driver and jConnect. IAnywhere JDBC driver requires the SQL Anywhere ODBC driver which is not case for the weather database, so the jConnect which communicates with SQL Anywhere using the TDS client/server protocol is chosen for the JDBC programs. JDBC program structure is like following:

- Create a Connection object

Calling a getConnection class method of the DriverManager class creates a Connection object, and establishes a connection with a database.

- Generate a statement object  
The Connection object generates a Statement object
- Pass a SQL statement

A SQL statement that executes within the database environment is pass-ed to the statement object. If the statement is a query, this action returns a ResultSet Object.

The ResultSet object contains the data returned from the SQL statement, but exposes it one row at a time.

- Loop over the rows of the ResultSet

The next () method of the ResultSet object performs two actions:

The current row advances one row and a Boolean value returns to indicate whether there is, in fact, a row to advance to.

- For each row, retrieve the values

Values are retrieved for each column in the ResultSet object by identifying either the name or position of the column. You can use the getData method to get the value from a column on the current row.

JDBC programs can obtain the weather data from database, but the problem is java applets are not able to present the data which directly are selected from database because it violates the security protocol of java applets. So File system is an intermediate connection between JDBC programs and Applet.

File System consists of several data files. They are all stored inside one directory so-called "DataFiles" on the web server and they all can be read and written.

Five kinds of Data files exist on the web server:

- Currentweatherdata file

Store the current weather data and rewrite every one minute

- Dayweatherdata file

Dayweatherdata files which are under the sub directory "DayDataFiles" of "DataFiles" should contain all measure parameters' data per hour in one day and the number of records in the file in the form of "day-month-yearweather.dat".

- Monthweatherdata files

Monthweatherdata files which are under the sub directory "MonthDataFiles" of "DataFiles" store all the measured parameters' data per day in one month and the number of records in the file in the form of "month-yearweather.dat"

- Yearweatherdata files

Yearweatherdata files which are under the sub directory "YearDataFiles" of "DataFiles" store all the measured parameters' data per month in one year and the number of records in the file in the form of "yearweather.dat".

➤ Contents files

Contents files list all the all files which exist on the server line by line.

### 4.3 Interactive GUI

Weather applets mean the graphical user interface which plot or display characteristic of measured parameters in the form of java applets.

Once the data files are ready on the server, applets are able to read these data via URL connection and present it in different manner. With the help of JFreeChart package, different measured parameters can be present in meaningful manner. For example, Thermometer figure is an appropriate choice for temperature display. It gives users an impression to know the current weather condition. On the other hand, if the user wants to see a temperature characteristic for one day, he needs to select the date of temperature and submit it on the website. A xyplot characteristic of temperature will be displayed in the form of applet afterwards.

One possible idea to gather user selection on a website is to construct an applet which provides several options for user such as the date, measured parameters selection check box, type of chart selection. After the user click submit button, all selection will be written to a file and transfer into file system, but the drawback of this solution is applet theoretically cannot write something to file except FTP connection. In order to transfer file via FTP connection, username and password must be provided in an applet which means the user can download applets and get to know the username and password of the server. That seriously violates the security of java applet.

Another solution is to combine user selection and chart into same applet on one web page, so that user can see the charts immediately after he clicked the plot button. The advantage of this solution is that user can see the charts directly after he selects the date, measured parameters and the type of chart.

### 4.4 Three-tier Architecture

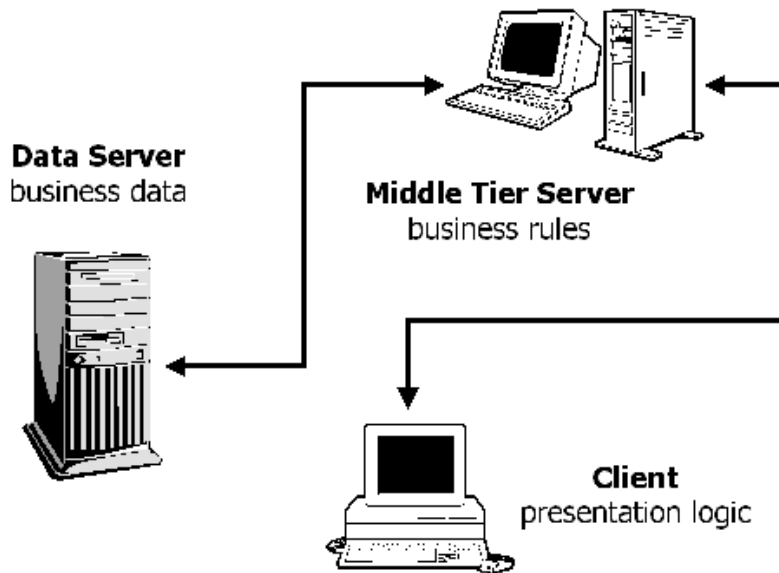
**Three-tier** is a client –server architecture in which the UI, functional process logic (“business rules”), computer data storage and data access are developed and maintained as independent modules, most often on separate platforms.

The three-tier model is considered to be software architecture and a software design pattern.

The three-tier architecture is intended to allow any of the three tiers to be upgraded or replaced independently as requirements or technology change. For example, a change of operating system in the presentation tier would only affect the UI code.

Typically, the UI runs on a desktop PC or workstation and uses a standard GUI, functional process logic may consist of one or more separate modules running on a workstation or application server, and an RDBMS on a database server.

Three-tier architecture (Figure 10) has the following three tiers:



*Figure.10 Three-Tier Architecture*

- Tier 1: the client contains the presentation logic, including simple control and user input validation. This application is also known as a thin client.
- Tier 2: the middle tier is also known as the application server, which provides the business processes logic and the data access.
- Tier 3: the data server provides the business data.

Weather station website is three tier architecture website. The bottom tier is the database server, which communicates with real weather station hardware via TCP/IP protocol. The middle layer is separated into two parts. One is JDBC java programs, which uses the standard SQL syntax in java to retrieve the data from weather database. Another one is the file system. File system contains different types of files such as measured value per days' data files, current weather data files etc. The purpose of this file system is to build a bridge between presentation logic and database server. The upper tier is the presentation logic, which display the weather data in a graphical manner. When a user browse the weather station website, it is always impressive that weather condition is shown graphically. In addition, as a dynamic website, the interaction between web browsers and website is also important because users likes to decide which kind of figures should be plotted and displayed in a website.

Figure 11 presents a UML deployment diagram that shows the physical relationship among the weather station website.

Even though the design described gives the impression of requiring a different machine for each tier, all tiers can be run in the same computer. This means the complete application is able to run in a single Linux or windows system with a graphical desktop, and it doesn't even have to be connected to the Net.

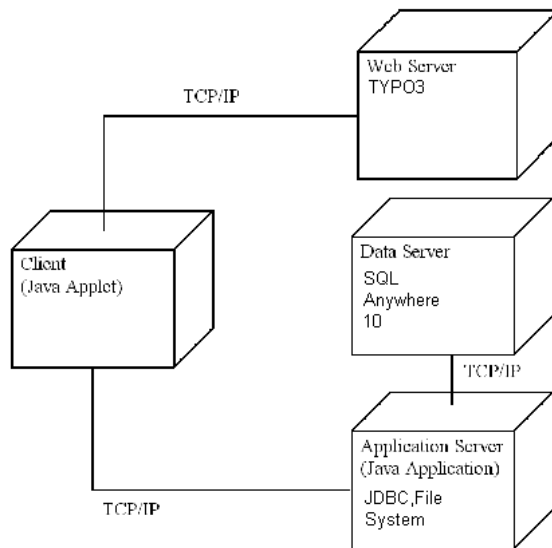


Figure.12 three tier architecture of Weather station

## 4.5 JAVA programs compile and execution on the Server

Every Java program needs to be transferred to web server via FTP connection such as Filezilla FTP client. The java applet program such as UserSelectionApplet needs to be compiled by “javac” command:

```
javac -classpath .:jfreechart-1.0.13.jar:jcommon-1.0.16.jar xxx.java.
```

The JDBC programs needs to be compiled by “javac” with jar file jconn2.jar command such as:

```
javac -classpath .:jconn2.jar xxx.java
```

Jfreechart1.0.13.jar and jcommon1.0.16.jar are the java packages from JFreeChart. Jconn2.jar is the JDBC package from SQL Anywhere 10. They all should be uploaded to the web server too.

Java applet needs to be deployed by manually coding the <applet> html tag. The following code is an example of wind direction applet using in the homepage of weather station.

```
<Applet ARCHIVE="jfreechart-1.0.13.jar,jcommon-1.0.16.jar"  
CODE="CurrentWindDirectionApplet.class" CODEBASE="//iweb.etech.haw-  
hamburg.de/wetter/JDBCconnection" width="200" height="250">
```

```
<PARAM NAME=LANGUAGE VALUE="Deutsch">
```

```
</Applet>
```

The attribute ARCHIVE describes one or more archives containing classes and other resource that will be “preloaded”. In this case, two jar files are included in order to plot the charts. “CODEBASE” attribute specifies the base URL of the applet which means the directory that contains the applet’s code. “CODE” attribute gives the name of file that contains the applet’s compiled Applet subclass. “PARAM” specifies the parameters name and value which could be got by Applets in the program.

JDBC programs should be packed into a jar file so that server just only executes one jar program at all. There are three step required to pack JDBC programs to jar file

- Modify manifest file, setting main-class which server needs to execute and add jconn2.jar to classpath of jar file
- Compile all JDBC programs and relevant java programs
- Create a JAR file by the following command:

```
Jar cfm JDBCMain.jar Manifest.txt JDBC programs.class
```

Once the jar file has been generated, this jar files needs to be executed by the server itself. The java command like following should be executed.

```
Java -jar directory/jdbc programs.jar
```

In the end, a cron job of linux OS should be started by the server administrator. Cron job is time based job scheduler in Unix-like computer operation systems. Cron enables users to schedule jobs (commands or shell scripts) to run automatically at a certain time or date. It is commonly used to automate system maintenance or administration. In this case above jar file should be executed by the cron-job.

## **5. Weather Station Website Design**

Weather station website consists of several sub web pages on TYPO3 system.

- Homepage
- Relevant information about hardware of weather station MWS6
- Weather database
- Contact form

➤ Imprint

**Homepage** provides graphics of current weather condition, different links to sub pages of weather station website and weather forecast widget which is associated with other weather website.

**Info** describes the overview of weather station including hardware MWS6 and general description of weather station web application

**Weather Database selection GUI** allows user to selection the date of chart for one day, one month and one year, number of measured parameters, and plot buttons corresponding to different types of chart.

**Contact Form** allows user to submit his questions and make contact with the web master if any problem occurs.

**Imprint** is the general information of the author of this weather station website.

Homepage is the main web page which contains the rest of web pages. They are all located on the tree from server on TYPO3.

## 5.1 Weather Station User Case Diagram

Weather station user case diagram in the unified Modeling language (UML) is a type of behavioral diagram defined by and created from a User- case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals, and dependencies between those user cases.

Weather station user case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted (Figure 13)



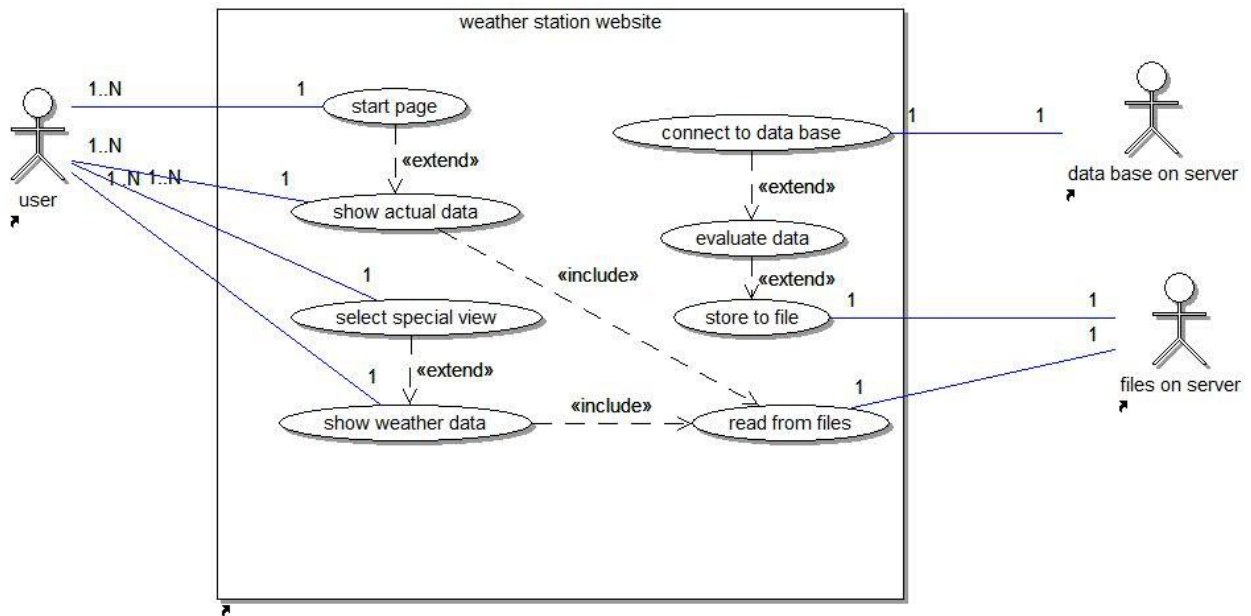


Figure 13 weather station use case diagram

There are three actors in the user case diagrams: users, database on server and files on server. Users whose browser has java virtual machine installed are able to see the actual data at start page in the form of Text or visualized graphics.

If user goes to the weather database sub page, the weather selection GUI will be shown. After user finishes his selection and click plot button, he will see the weather charts according to the number of measured parameter and type of chart user has been selected.

The weather data in the form of graphical view comes from files on the server.

On the server side, server executes four different JDBC programs in four different threads.

- CurrentWeatherDataJDBC program

CurrentWeatherDataJDBC program retrieve top 1 record from weather database every one minute.

- DayWeatherDataJDBC program

DayWeatherDataJDBC program retrieve 24 mean weather values in one day every one hour.

- MonthWeatherDataJDBC program

MonthWeatherDataJDBC retrieve one month mean weather values every one day.

- YearWeatherDataJDBC program

YearWeatherDataJDBC program retrieve one year mean weather values every month.

All the JDBC programs connect to database, evaluate data and the retrieved data will be written into corresponding files and store into different directory on the server. The contents file list all the existed files on the server. The applet reads the contents file and provide the user which day the user can choose and see the chart afterwards.

## 5.2 Weather Station Class Diagram

Class diagram in the UML is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, and the relationships between the classes.

The class diagram is the main building block in object oriented modeling. They are being used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the model into programming code. The classes in a class diagram represent both the main objects and or interactions in the application and objects to be programmed. In the class diagram these classes are represented with boxes which contain three parts.

- The upper part holds the name of the class
- The middle part contains the attributes of the class, and
- The bottom part gives the methods or operations the class can take of undertake

Weather Station class diagram (Figure 14) consists of two kinds of class diagrams:

- Class diagrams of JDBC programs

These diagrams include CurrentWeatherDataJDBC, DayWeatherDataJDBC, MonthWeatherDataJDBC, YearWeatherDataJDBC and their corresponding thread classes which extend Thread.

JDBCMain class contains all four instances of JDBC programs and they run in separate thread.

- Applet class diagrams

CurrentWeatherApplet class is shown at the homepage of weather station website. UserSelectionApplet class is weather database selection GUI. The user selection and weather charts display will occur in the same applet. The other applet

like CurrentBarometerApplet is the graphical view of current weather condition on Homepage. They all implement the interface “WeatherValue”.

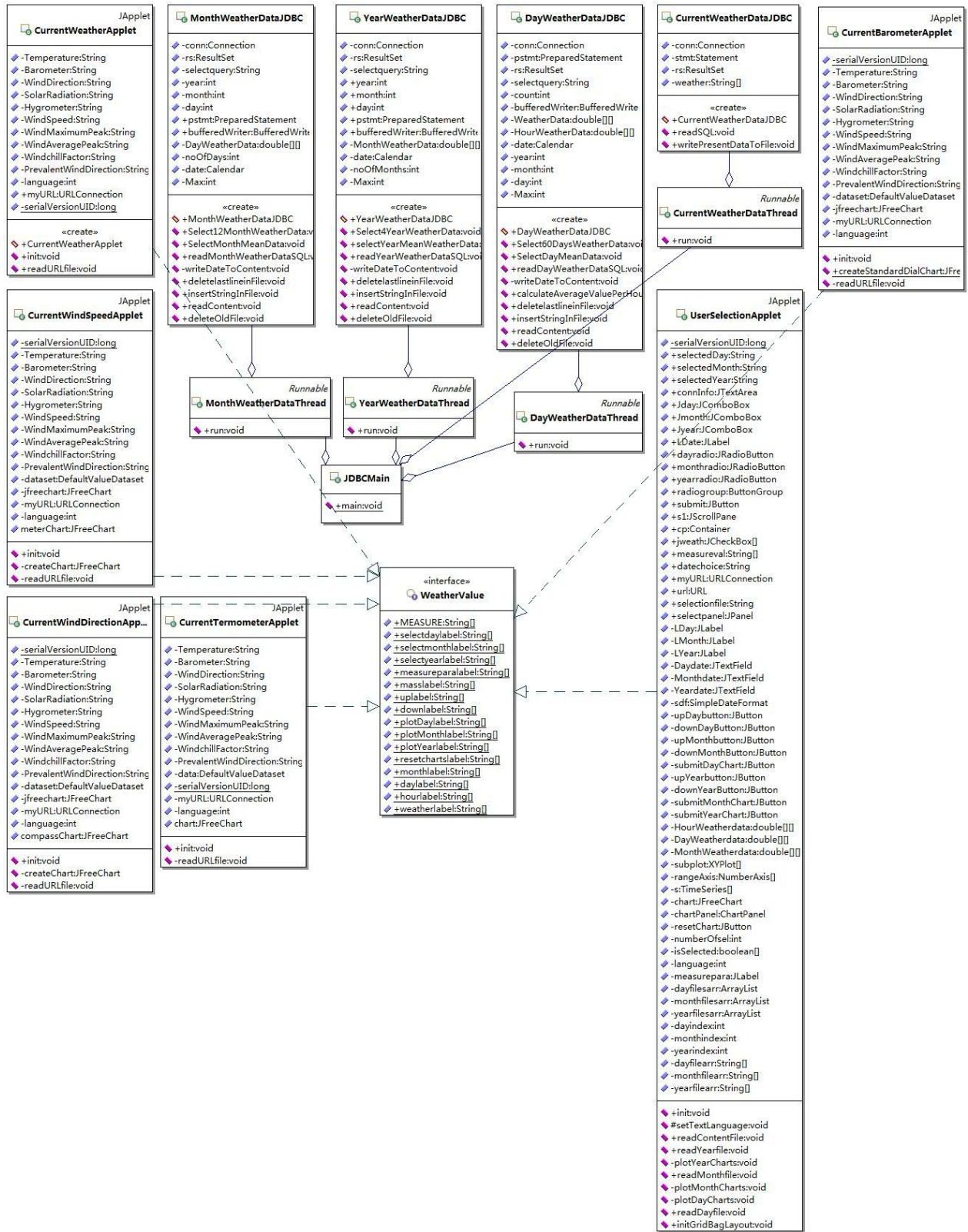


Figure 14 class diagram of weather station

### 5.3 Weather Station Homepage Design

Weather station homepage consist of CurrentWeatherApplet which displays the current weather condition, the sitemap of weather station and several visualize graphics for specific measured parameters. CurrentWeatherDataJDBC program generate the current weather data file which is read by the CurrentWeatherApplet. The main method in CurrentWeatherDataJDBC is readSQL() method. Figure15 is the sequence diagram of readSQL() method.

readSQL() method basically is the standard procedure how to retrieve the data from weather database and write the data to file system. Register SQL Anywhere database driver, get connection to the weather DB, create prepared SQL statement, executed select query and obtain the result set. The for loop indicates there are ten parameters' data which needs to save in an array by getString() of result set. After the data has been retrieved from weather DB, all connection, statement and resultset must be close. Next step is to write the data which is stored in an array to a file. BufferedWriter and FileWriter are used in this case because BufferedWriter has a simple method write(String) which write all data to "currentweatherdata.dat" file line by line. When File is finished, the bufferedwriter needs to be flushed and closed.

In order to update the current weather condition every one minute, Homepage must be refreshed every one minutes. The following meta tag needs to be added inside the homepage.

```
<META HTTP-EQUIV="Refresh" CONTENT="60";URL=http://iweb.etch.haw-hamburg.de/yamlt3/index.php?id=398">
```

The url address is the address of homepage. It automatically refreshes the homepage every 60 seconds.

The other applet such as CurrentBarometerApplet, or CurrentTermometerApplet are the graphical view of current weather condition. They all read the data from "currentweatherdata.dat" file.

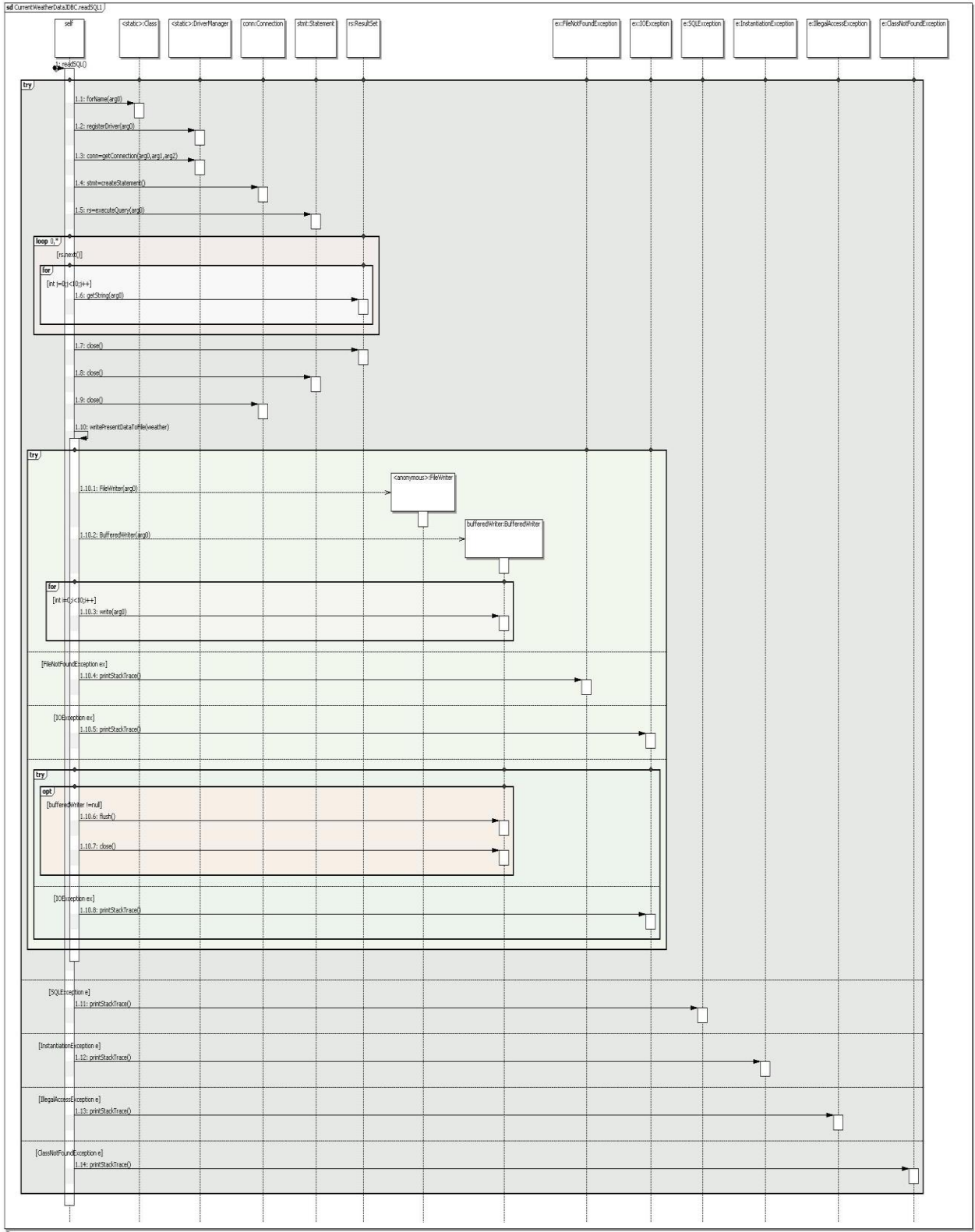


Figure 15 sequence diagram of readSQL () method

## 5.4 Weather Station Database GUI Design

Next important Design for weather station is the database GUI. It provides the user several options to select.

- Date selection for last 60 days
- Date selection for last 12 months
- Date selection for all years
- Measured parameters selection
- Four plot button including a reset button

The layout of this GUI uses the Border layout of java swing class. The date selection locates on the top of the GUI, the measured parameters selection is on the left side of GUI, buttons are on the bottom of GUI. Because the limit space of website, the maximal select option for measured parameter is four, so that user can see four different charts on one page simultaneously after one button is clicked.

There are several methods in the “UserSelectionApplet”.

- readContentFile()

readContentFile() method read the contents file which is the date of data files in the form of “day-month-year” and store them in a ArrayList. The textfield of Date selection presents the content in that contents file. The up and down button is like the pointer to point the next or previous line in that contents file

- readDay(month,year)file()

This kind of method reads the data from the data files and stores the data in a 2D array. First of all, the data files on the server must be treated as URL resource otherwise it will violate the security protocol of applet. Secondly, the “readline()” method of the class bufferreader is used in this case and every line represents the same record inserted into table in weather database except they should be spilt by white space.

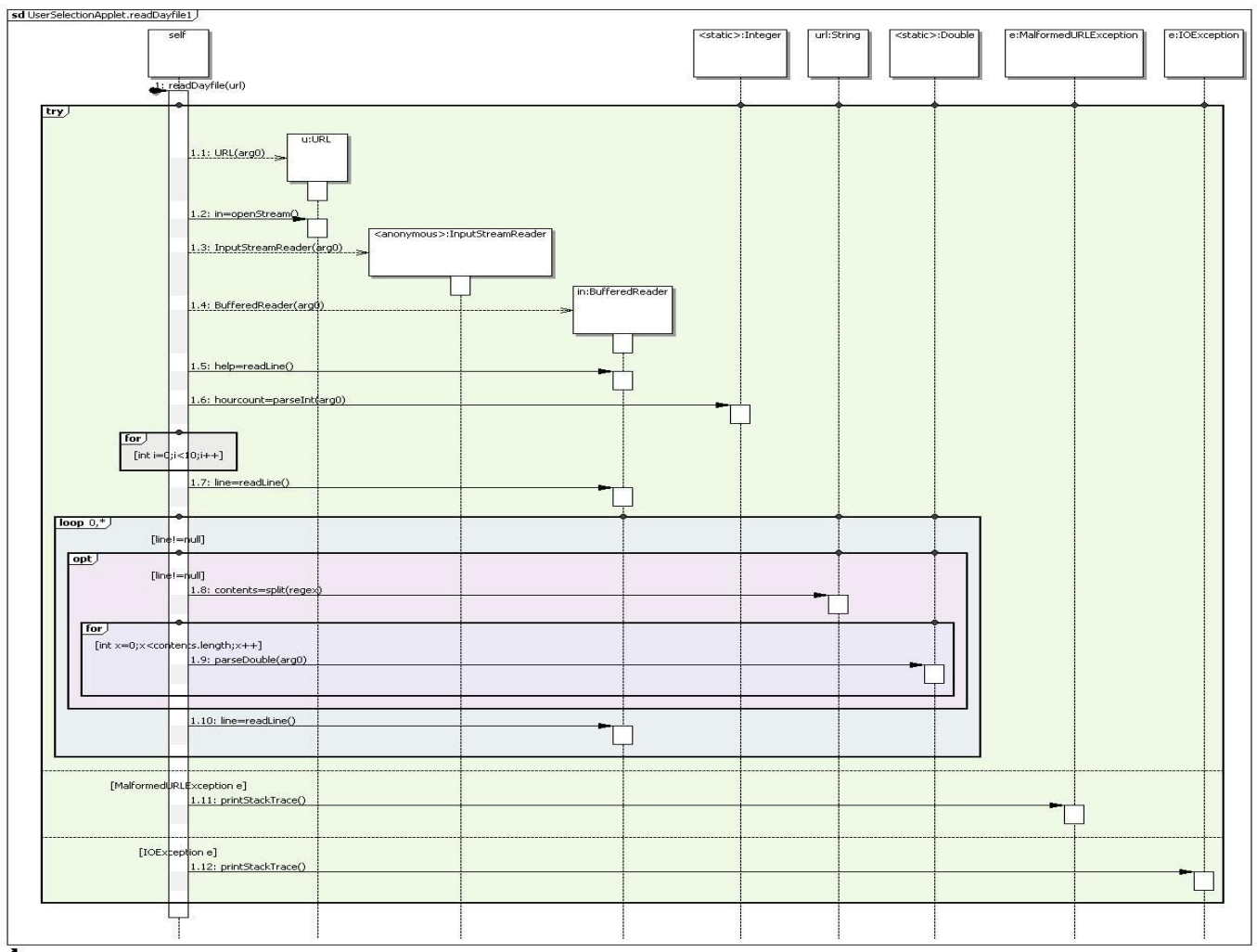
- PlotDay(Month,Year)Charts()

Those are different plot methods which generate different chart. The Combined-DomainXYPlot class from JFreeChart framework is used. Depending on the number of measured parameter user selected, CombinedDomainXYPlot can generate the same number of charts with same domain axis. The plot method will only be activated if the plot button has been clicked and after the reset button has been clicked, the whole chart panel will show blank.

- setTextLanguage()

In order to give the user different possible language selections, every label, textfield, checkbox and buttons need to be provided two languages version: English, and German. They are in an array of string which has two elements: one is the English version string and another one is German version string. In applet, first of all, applet gets parameter "language" which is specified on the website. A variable "language" in the applet is used as the flag. If language is set to one, English version is chosen, all the GUI elements pick up the first string from array, if language is set to zero, all the GUI elements pick up the second string from array

A sequence diagram readDayFile is shown Figure 16. Because applet won't allow FileReader, the file will be treated as URL on the server. The BufferedReader is ready to read the content line by line after URLstream is open. First of all, the variable "hourcount" is to save the count of line in the data file. The data are read by readline() buffered Writer and stored in a two dimension array "HourWeatherData". The size of array depends on the hourcount read in the first line.



*Figure 16 sequence diagram of readDayFile method*

## 5.5 JDBC programs Design

JDBC programs are the server side programs which are running by the server via cron job in separate threads all the time. Besides the CurrentWeatherJDBC program, Day, month, year WeatherDataJDBC programs have a contents file. When a new day comes, the server will select and calculate mean value per hour of all measured parameters and write them into a new file named as “day-month-yearweather.dat” (DayWeatherDataJDBC program). The string “day-month-year” which represents current day will also be inserted into the contents file at the first line. At mean time, the last line which represents the oldest data file will be erased in the contents file and on the server. The reason is that the number of data files maintains a fixed number and in case, server stop running in several days, the applet won’t read the date from contents file which the data files did not be generated.

Server only generates current day’s data files one time every one hour for DayWeatherDataJDBC, every day for MonthWeatherDataJDBC, every month YearWeatherDataJDBC and the old data files are already on the server. The benefit of the way is to reduce the time consumption run by the server because server needs to run four threads in parallel and CurrentWeatherDataJDBC needs to be run every minute.

In order to prevent the lost of all data files on the server, the server needs to regenerate all the old data files if the data file directory is empty.

Figure 17 is the sequence diagram of the method Select60DaysWeatherData from DayWeatherJDBC program. This method start in a for loop. The current date is obtained by getInstance of Calendar class. Date subtracts by 1 after each iterator until 60 days, so that 60 old Days weather data is generated and stored on the server.

ReadDayWeatherDataSQL method is called inside for loop with year, month, day parameters. This method follows standard JDBC protocol: register sql anywhere 10 JDBC driver, establish connection and prepare the sql statement. Be aware of one thing that the size of two dimension array which store all the weather data in one day is dynamic. It depends on the count of records being inserted into database, so the select sql statement needs to be executed twice. Once it retrieves the count of records in the database, second time it select the weather data in one day from database. After the data are in the 2D array “WeatherData”, the method calculateAverageValuePerHour is called, so that a new 2D array called HourWeatherData contains the mean value of all weather data per hour. The bufferedWriter is an instance of BufferedWriter It has the write method which writes the content of 2D HourWeatherData array to a file line by line. Each data is separated by white space. The file is named by its date in the form of “year-month-dayweather.dat”.



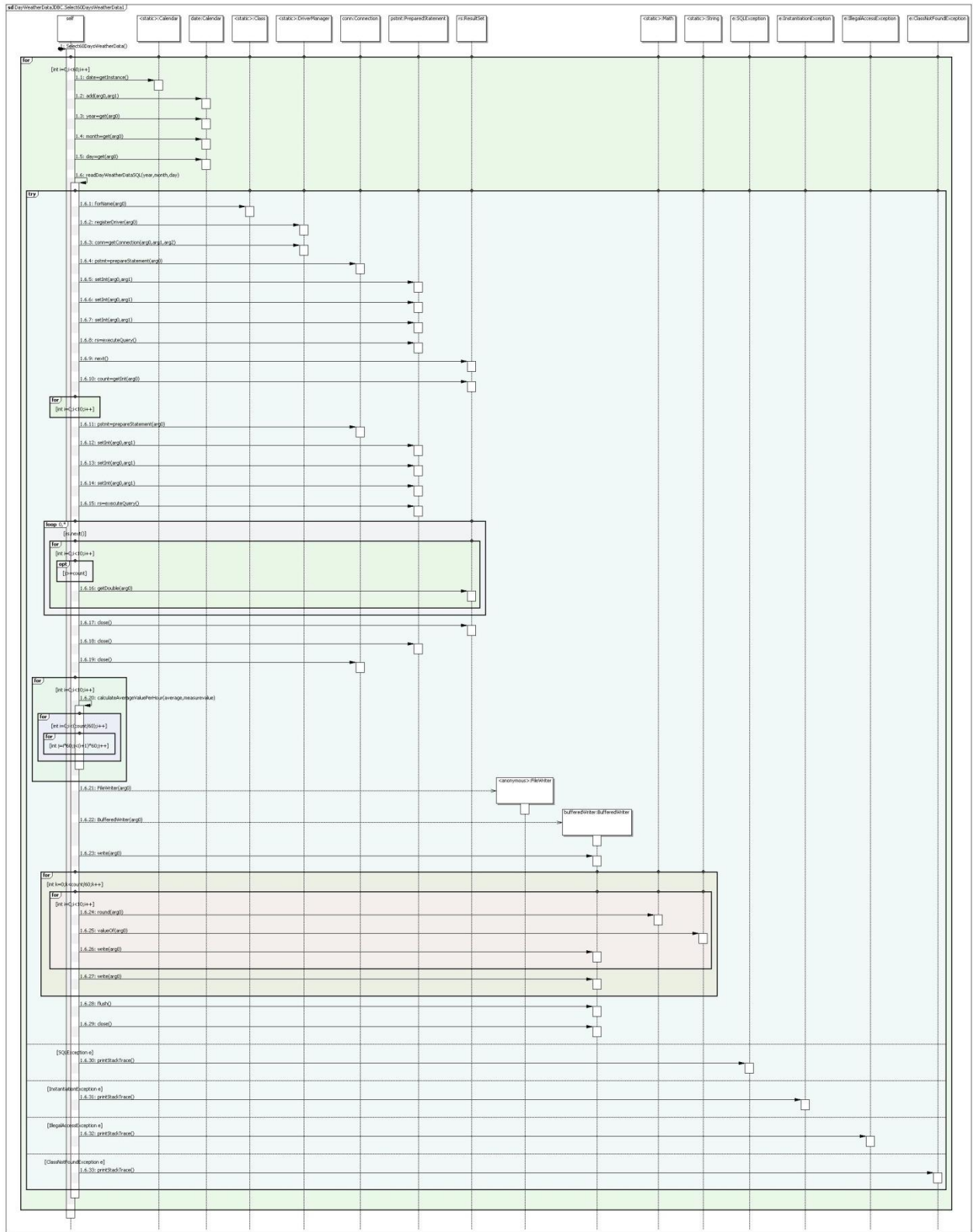


Figure 17 sequence diagram of `select60DaysWeatherData`

This method will be only called when the data file directory is empty. All data files have been deleted in case.

Figure 18 is the sequence diagram of `readcontent(String filepath,int day, int month, int year)` method in `DayWeatherDataJDBC` program. It reads the first line in contents file and save it in a variable “temp”. Afterwards, it read all the records in the contents file line by line and save them an arraylist. If a new day comes, the string of “day-month-year” is different with the current first line of contents file which represents the day before that new day. Next the capacity of contents file need to be check, if the number of date records has reached the maximum which set it to 60 in this case, the last line record must be erased and the oldest file also. The current date record will be inserted into first line of contents file. If not reach the maximum capacity of contents file, the oldest data file and record won't be deleted.

The method “`void insertStringinFile(File in File, int lineno, String lineToBeInserted)`” is the one insert line to contents file. The key point of this method is to create a temporary contents file “`contents.tmp`”. Read the contents file first, write the string to line which line number is known as parameter and write the other lines which read from original contents file. After that delete the original contents File, rename the temporary file to contents file.

The method “`void deletelastlineinFile(String filepath)`” is more or less similar to the insert method. One needs to read the contents file except the last line, delete the original contents file and rename the temporary file to original contents file back.

The method “`deleteOldFile(String file)`” just check if the file exist on the directory on the server, if file is there ,delete that file.

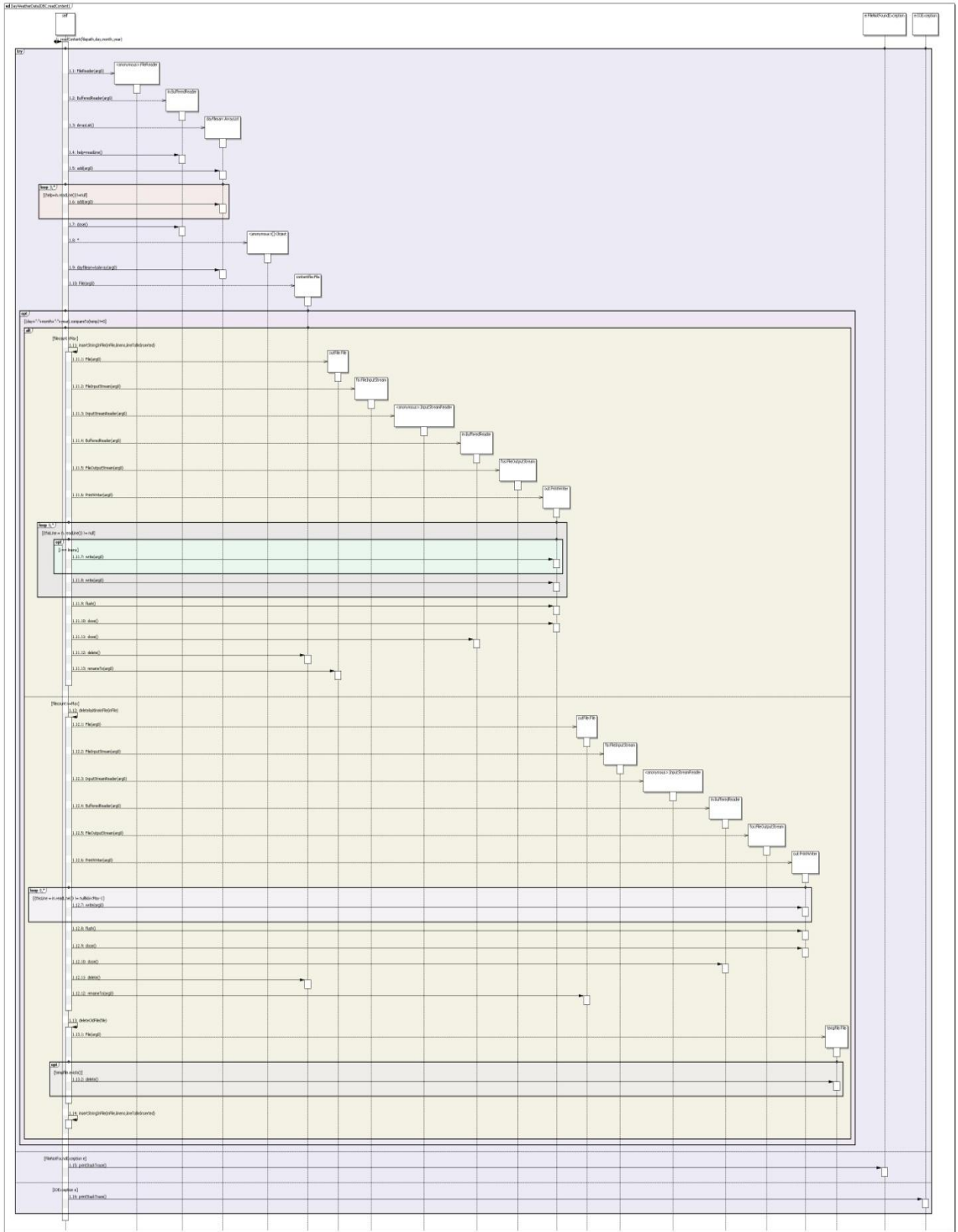


Figure 18 sequence diagram of readContent method

## 6. Weather Station Website Results

Figure 19 is the homepage of weather station. The left side of homepage is the current weather condition applet in the form of text. The right side is graphical display of temperature and barometer. The German and English flag indicates this web page is available in two different languages. User can click the flag to switch the language they prefer to browser.

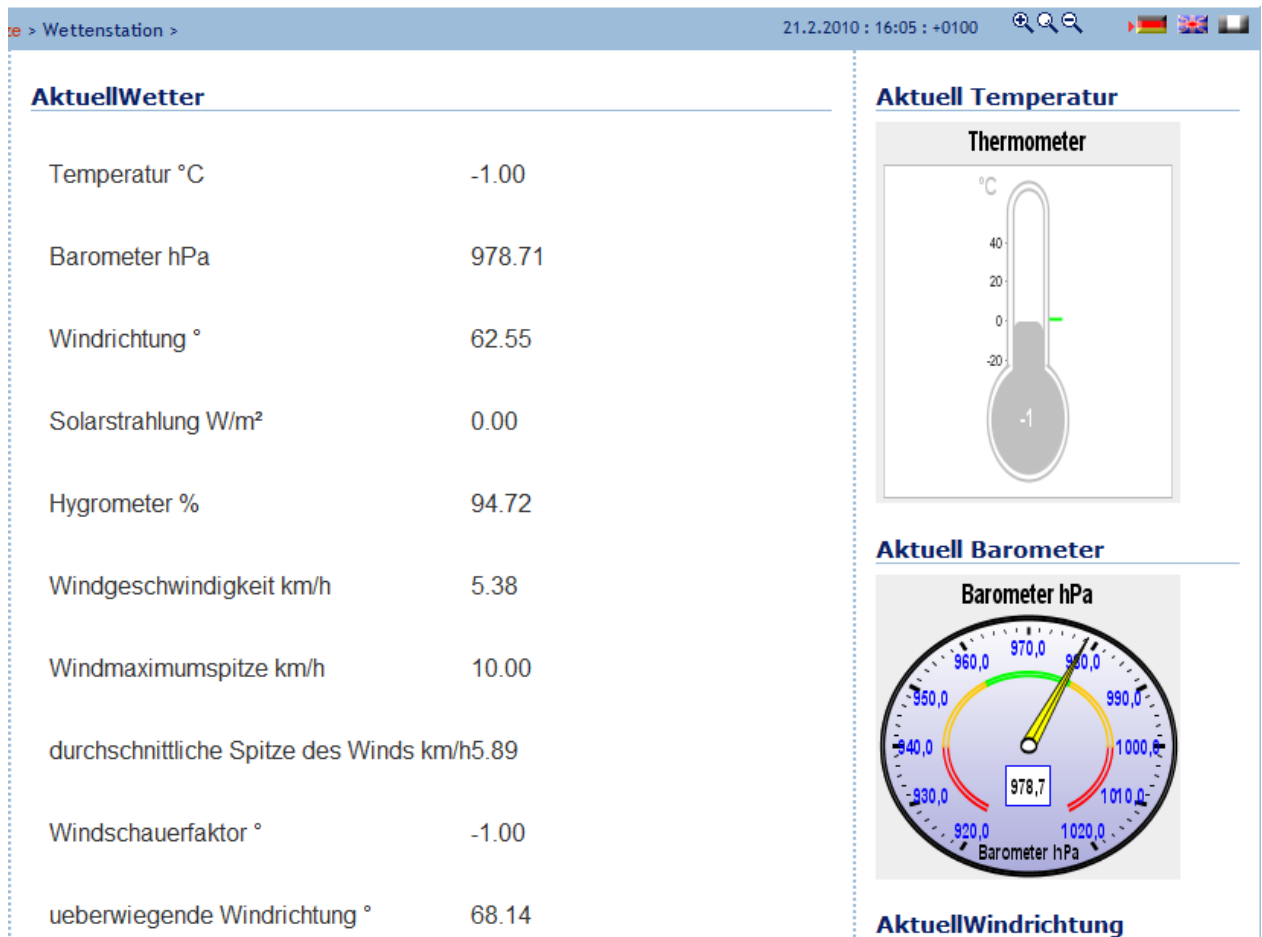
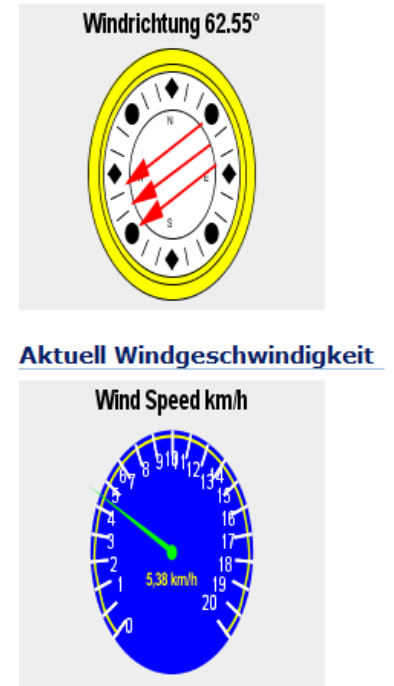


Figure 19 homepage of weather station

The weather station links is the sitemap of weather station web site. User can click different link button to different sub pages. Another speed dials will represent the current wind direction and wind speed (Figure 20).

### Wetterstation Sitemaps

- Wetterstation MWS 6 Information->[Link](#)
- Wetterstation Datenbank GUI ->[Link](#)
- Kontakt -> [Link](#)
- Impressum -> [Link](#)



*Figure 20 sitemap and forecast widget of weather station*

When user clicks the link to weather database, user can see the current and old weather data characteristic per day, per month, and per year. The default date for all charts is today, this month and this year. Figure 21 is the weather database GUI for user. The maximum number of measured parameter selection is 4. The user won't be able to select more measured parameters to display in one chart simultaneously due to limit size of web page.

**Datenbank GUI** (Blockieren)

Datum fuer Tagesdiagramme vorwaehlen	<input type="text" value="18-2-2010"/>	<input type="button" value="oben"/>	<input type="button" value="unten"/>
Datum fuer Monatsdiagramme vorwaehlen	<input type="text" value="2-2010"/>	<input type="button" value="oben"/>	<input type="button" value="unten"/>
Datum fuer Jahrdiagramme vorwaehlen	<input type="text" value="2010"/>	<input type="button" value="oben"/>	<input type="button" value="unten"/>

**gemessene Wetterparameter bitte waehlen**

- Temperatur °C
- Barometer hPa
- Windrichtung °
- Solarstrahlung W/m<sup>2</sup>
- Hygrometer %
- Windgeschwindigkeit km/h
- Windmaximumspitze km/h
- durchschnittliche Spitze des Winds km/h
- Windschauerfaktor °
- ueberwiegende Windrichtung °

<input type="button" value="Tagesdiagramme erzeugen"/>	<input type="button" value="Monatsdiagramme erzeugen"/>	<input type="button" value="Jahrdiagramme erzeugen"/>	<input type="button" value="Diagramme zurueckstellen"/>
--	---	---	---

*Figure 21 weather database GUI*

For example If user wants to see the 1-2-2010's temperature and barometer characteristic charts, user needs to click "down" button of first line and select Temperature and Barometer options on the left side and then click "Tagesdiagramme erzeugen" button. Figure 22 is the temperature chart on 1-2-2010. Again, the web page can be translated into English and German.

## Datenbank GUI

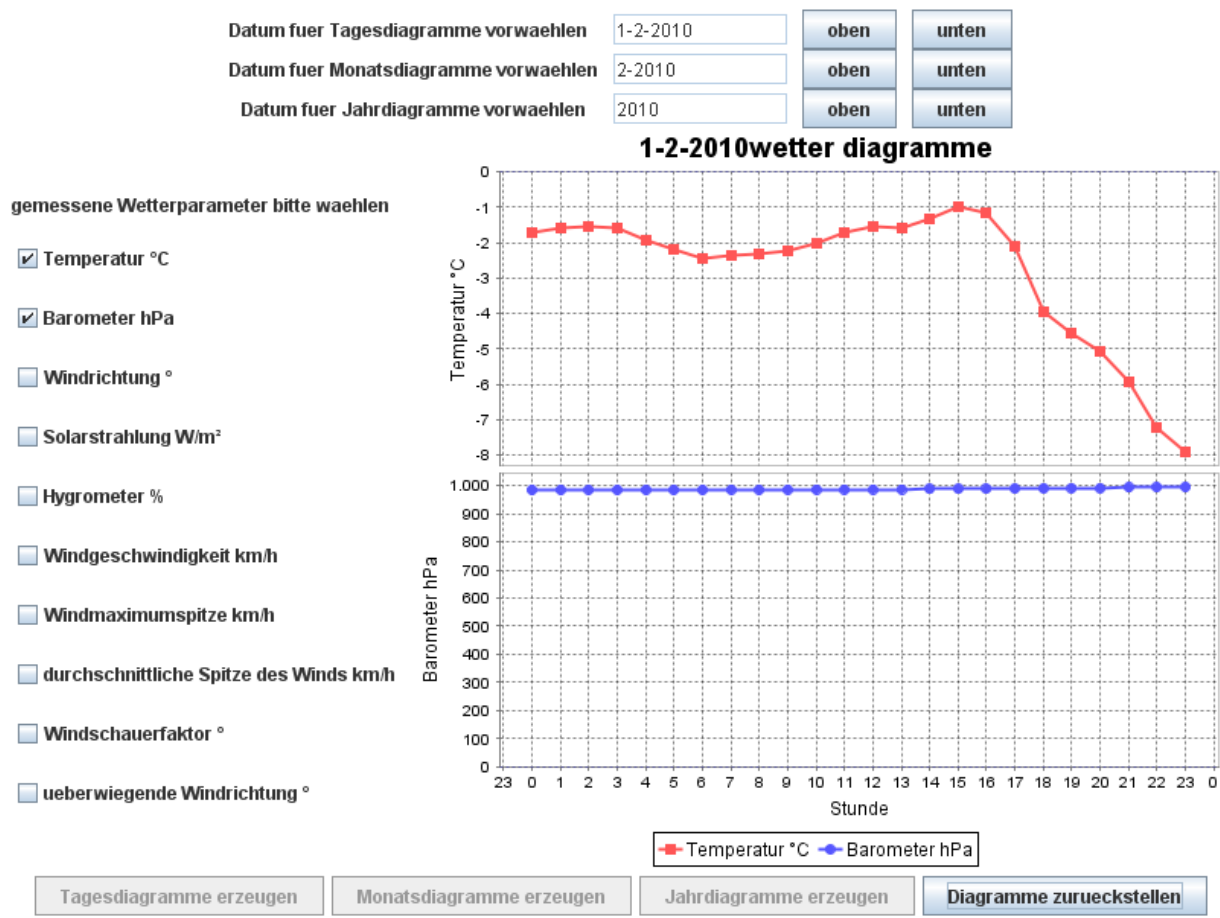


Figure 22 Demo for Temperature and Barometer chart on 1-2-2010

After the chart has been plotted, the “Tagesdiagramme erzeugen” button is invisible. The “ResetChart” button is visible, which means the chart panel must be clear up before another chart is plotted.

If user wants to know the detail about the hardware of weather station hardware MWS6, just goes back to home page and click the link “MWS6Info” and the general description

of MWS 6 will be display at the next page (Figure 23)

#### MWS6 Information

Eine Wetterstation ist eine Anlage mit Instrumenten und Ausrüstung für das Beobachten der atmosphärischen Bedingungen, um Informationen für Wettervorhersagen zur Verfügung zu stellen und das Wetter und das Klima zu studieren. Die genommenen Maße umfassen Temperatur, Barometer, Feuchtigkeit, Windgeschwindigkeit, Windrichtung und Niederschlagsmengen. Typische Wetterstationen haben einige Instrumente einschließlich Thermometer, Barometer, Hygrometer, Anemometer, Windschaukel, Regenlehre.

Die Wetterstation, die auf dem Dach der Universität gelegen ist, ist MWS 6, das von Reinhardt - Testsysteme gekommen wird. Die Aussicht von MWS 6 ist Tabelle 2 und der interne Entwurf ist Tabelle 1. MWS 6 ist ein Mikroprozessor - Wetterstation mit 6 Sensoren für 6 messende Parameter, wie folgend:

- Temperatur ° Celsius
- atmosphärischer Druck hectoPascal
- Feuchtigkeit %
- Windgeschwindigkeit km/h
- Wind-Richtung °
- Sonnenenergie Watt/m<sup>2</sup>

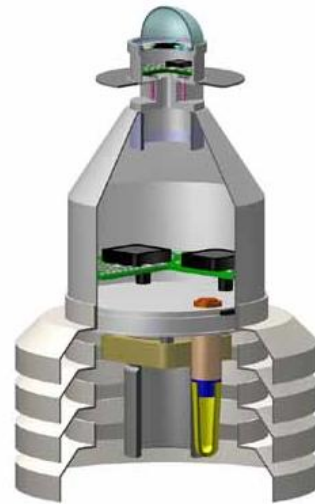
Die MWS 6 Wetterstation hat eine spezielle Methode des Messens der Windrichtung und der Windgeschwindigkeit (0 bis 50km/h): Drei Windgeschwindigkeit-Sensoren notieren die notwendigen Werte.

Maximale Spitze, durchschnittliche Spitze des Winds, Windschauerfaktor wickeln und überwiegende Windrichtung werden gemessen. Ein pyranometer für das Messen der globalen Strahlung (Sonnenenergie) ist als Temperatur, relative Luftfeuchtigkeit, Taupunkt und Luftdruck Standard. Daten werden über RS232 - Schnittstelle übertragen. Mit IP-Modul ist eine Datenübertragung über ein vorhandenes Netz vorhanden.

#### WeatherStationDevice



mws6



MWS-MC

Figure 23 Information of MWS 6 weather station



## 7. Analysis of Temperature Extrapolation Algorithms

This chapter emphasizes on analysis of temperature extrapolation algorithms. Different forecasting models are summarized and introduced.

The second part of this chapter is to explain the reason why Henderson Trend filter and double smoothing algorithms are used to design Temperature of one day forecasting model is introduced and compared with other algorithms. The design of forecasting program is the second part of chapter. This program is constructed by java language with the help of JFreeChart package. The third part of this chapter contains the result of two algorithms in the form of graphic and table comparing with the original temperature.

### 7.1 Forecasting methods

Forecasting is the process of making statements about events whose actual outcomes have not yet been observed. A commonplace example might be estimation of the expected value for some variable of interest at some specific future date. Prediction is a similar, but more general term. Both might refer to formal statistical methods employing: time series, cross-sectional or longitudinal data or alternatively to less formal judgmental methods.

The categories of forecasting methods are following:

➤ Time series methods

Time series methods use historical data as the basis of estimating future outcomes

- Moving Average
- Weighted moving average
- Exponential smoothing
- Extrapolation
- Linear predication
- Trend estimation
- Growth curve

➤ Causal / econometric methods

Some forecasting methods use the assumption that it is possible to identify the underlying factors that might influence the variable that is being forecast. For example, sales of umbrellas might be associated with weather condition. If the causes are understood, projections of the influencing variable can be made and used in the forecast.

- Regression analysis using linear regression or non-linear regression
- Autoregressive moving average (ARMA)

- Autoregressive integrated moving average (ARIMA)
- Econometrics
- Judgmental methods

Judgmental forecasting methods incorporate intuitive judgements, opinions and subjective probability estimates.

- Composite forecasts
- Surveys
- Delphi method
- Scenario method
- Technology forecasting
- Forecast by analog

The vital issue of forecasting model is the forecasting accuracy. The forecast error is the difference between the actual value and the forecast value for the corresponding period.

$$E_t = Y_t - F_t$$

Where E is the forecast error at period t, Y is the actual value at period t, and F is the forecast for period t. Measures of aggregate error.

Mean Absolute Error (MAE):  $MAE = \frac{\sum_{t=1}^N |E_t|}{N}$

Mean squared error (MSE)  $MSE = \frac{\sum_{t=1}^N E_t^2}{N}$

Forecasting has many applications in many situations such as supply chain management, weather forecasting, Flood forecasting, transport planning and so on.

Temperature extrapolation is one kind of weather forecasting that is the application of science and technology to predict the state of the atmosphere for a future time and a given location. Weather forecasting are made by collecting quantitative data about the current state of the atmosphere and using scientific understanding of atmospheric processes to project how the atmosphere will evolve.

Forecast models are now used to determine future conditions. Human input is still required to pick the best possible forecast model to base the forecast upon, which involves pattern recognition skills, teleconnections, knowledge of model performance, and knowledge of model biases.

Four steps consist of forecasts creation of models:

- Data collection
- Data assimilation and analysis

- Numerical weather prediction
- Model output post processing

The algorithms of temperature extrapolation are developed in next 2 sections.

They are Henderson trend filter algorithm which extends the one of the time series method trend estimated.

Another one is the double exponential smoothing which is a technique that can be applied to time series data, either to produce smoothed data for presentation, or to make forecast.

## **7.2 The principle of Trend Filter**

The idea of trend filter or so-called trend moving average comes from trend estimation.

Trend estimation is a statistical technique to aid interpretation of data. When a series of measurements of a process are treated as a time series, then the application of trend estimation can be used to make and justify statements about trends in the data. Using trend estimation, it is possible to construct a model which is independent of anything known about the physics of a process of an incompletely understood physical system. This model can then be used to explain the behavior of a measurement. In particular, it may be useful to determine if measurements exhibit an increasing or decreasing trend, which is statistically distinguished from random behavior. The trend of the daily average Temperatures at a given location, from winter to summer, or the trend in a global temperature series over the last 100 years.

Trend filter in the case of daily temperature estimation is an algorithm that preliminarily looks at recent temperature behavior and determines whether the trend is up, down or neutral.

It is not essential that every approach have a separate trend filter. Some methods, such as moving averages, incorporate a trend indicator into the entry technique. Other tries to predict an imminent change in trend and are therefore entering when the trend is against them at the time.

Trend is only relevant in conjunction with a particular time frame. Once the time frame is determined, the daily temperature is either moving up or moving down during time period. In order to create a simple trend indicator, there is one intriguing issue to be considered. Which temperature components should be used to construct the indicator?

The Henderson trend filter will explain how to construct a trend indicator in next section.

## 7.3 Henderson Trend Filter for Temperature Extrapolation

The Henderson filters (Henderson Moving average) are a widely used set of trend filters. They were initially derived by Robert Henderson (1916) for use in actual work. They are commonly used in time series analysis to smooth seasonally adjusted estimates in order to generate a trend estimate.

The requirement used by Henderson to derive the filters was that they must follow a local cubic polynomial without distortion. The Henderson filters are derived by minimizing the sum of squares of the third difference of the moving average series. Henderson's criteria ensures that when the Henderson filters are applied to third degree polynomials, the resulting smoothed output will fit exactly on these parabolas. The Henderson filters are suitable for economic time series as they allow the cycles typical of the trend to pass through unchanged. They also have the property that they will eliminate almost all the irregular variations that are of very short frequencies of six months or less.

The filter weights applied in the middle of a time series are symmetric, while the end filter weights are asymmetric. This is due to the end problem where at the start and end of the series there is no enough data on either side of the data points to generate symmetric filter weights.

In this case, the filter weights suited to the temperature data should be asymmetric because the aim is to extrapolate the temperature data based on the known value.

There are a number of ways to derive the Henderson symmetric filter weights and the Henderson asymmetric filter weights.

The 13 term Henderson trend filter is used for temperature data estimation because of its good property for early turning point detection.

### 7.3.1 13-term Henderson Trend-cycle Filter

The 13-term Henderson filter as its variants, 9 and 23 term, are based on summation formulas mainly used by actuaries. The basic principle for the summation formula is the combination of operations of differencing and summation in such a manner that when differencing above a certain order is ignored, they will reproduce the functions operated on. The merit of this procedure is that the smoothed values thus obtained are functions of a large number of observations whose errors, to a considerable extent, cancel out. These filters have the properties that, when fitted to second or third degree parabolas, their output will fall exactly on those parabolas, when fitted to stochastic data; they will give smoother results than can be obtained from weights which give the middle point of a second degree parabola fitted by least squares. Recognition of the fact that the smoothness of the resulting filtering depends on the smoothness of the weight diagram

led Henderson to develop a formula which makes the sum of squares of the third differences of the smoothed series a minimum for any numbers of terms.

Henderson's starting point was the requirement that the filter should reproduce a cubic polynomial trend without distortion. Henderson proved that three alternative smoothing criteria give the same formula, as shown explicitly by Kenny and Durbin (1982) and Gray and Thomson (1996):

- Minimization of the variance of the third differences of the series defined by the application of the moving average.
- Minimization of the sum of squares of the third differences of the coefficients of the moving average formula
- Fitting a cubic polynomial by weighted least squares, where the weights are chosen as to minimize the sum of squares of their third differences.

The problem is one of fitting a cubic trend by weighted least squares to the observations  $y_{t+j}$ ,  $j=-m, \dots, m$ , the value of the fitted function at  $j=0$  being taken as the smoothed observation  $\hat{g}_t$ . Representing the weight assigned to the residuals from the local polynomial regression by  $W_j$ ,  $j=-m, \dots, m$ , where  $W_j=W_{-j}$ , the problem is the minimization of

$$\sum_{j=-m}^m W_j [y_{t+j} - a_0 - a_1j - a_2j^2 - a_3j^3]^2$$

where the solution for the constant  $a_0$  is the smoothed observation  $\hat{g}_t$ .

Henderson showed that  $\hat{g}_t$  is given by

$$\sum_{j=-m}^m \phi(j)W_j y_{t+j} = \sum_{j=-m}^m W_j y_{t+j}$$

Where  $\phi(j)$  a cubic polynomial is whose coefficients have the property that the smoother reproduces the data if they follow a cubic. Henderson also proved the converse: if the coefficients of a cubic-reproducing summation formula  $\{W_j, j=-m, \dots, m\}$  do not change their sign more than three times within the filter span, then the formula can be represented as a local cubic smoother with weights  $W_j > 0$  and a cubic polynomial  $\phi(j)$ , such that  $\phi(j)W_j = w_j$ . Henderson measured the amount of smoothing of the input series by  $\sum(\Delta^3 y_t)^2$  or equivalently by the sum of squares of the third differences of the weight diagram  $\sum(\Delta^3 w_j)^2$ . The solution is that resulting from the minimization of a cubic polynomial function by weighted least squares with

$$W_j \propto \{(m+1)^2 - j^2\}\{(m+2)^2 - j^2\}\{(m+3)^2 - j^2\}$$

$\propto$  means  $W_j$  is proportional to a cubic polynomial function. As the weighting penalty function of criterion:

$$\hat{g}_t(0) = \hat{g}_t = \sum_{j=-m}^m w_j y_{t+j}$$

If the filter length is  $2m-3$ , Henderson showed that the general expression for the weight diagram  $\{W_j, j= -m, \dots, m\}$  is:

$$W_j = \frac{315[(m-1)^2 - j^2](m^2 - j^2)[(m+1)^2 - j^2][3m^2 - 16 - 11j^2]}{8m(m^2 - 1)(4m^2 - 1)(4m^2 - 9)(4m^2 - 25)}$$

To derive a set of 13 weights from above equation, 8 is substituted for  $m$  and the values are obtained for each  $j$  from  $-6$  to  $6$ . The Henderson 13-term trend –cycle filter is thus given by,

$$H_{13}(B) = -.019B^{-6} - .028B^{-5} + .00B^{-4} + .065B^{-3} + .147B^{-2} + .214B^{-1} + .24B^{-0} + .214B + .147B^2 + .065B^3 + .00B^4 - .028B^5 - .019B^6$$

Where  $B$  is the backshift or lag operator defined by  $B^m y_t = y_{t-m}$  and  $B^0 = 1$ .

The calculation of the weights of the asymmetric Henderson filter is based on the minimization of the mean squared revision (MSR) between the final estimates (obtained by the application of the symmetric filter) and the preliminary estimate (obtained by the application of an asymmetric filter) subject to the constraint that the sum of the weights is equal to one. The assumption made is that at the end of the series the seasonally adjusted values are equal to a linear trend-cycle plus a purely random irregular  $\varepsilon_t$  such that  $NID \sim (0, \sigma_a^2)$ . The equation used is:

$$E[r_t^{(i,m)^2}] = c_1^2 \left( t - \sum_{j=-i}^m h_{ij}(t-j)^2 \right) + \sigma_a^2 \sum_{j=-m}^m (h_{mj} - h_{ij})^2$$

where  $h_{mj}$  and  $h_{ij}$  are the weights of the symmetric (central) filter and the asymmetric filters, respectively;  $h_{ij} = 0$  for  $j=-m, \dots, -i-1$ ,  $c_1$  is the slope of the line and  $\sigma_a^2$  denotes the noise variance. There is a relation between  $c_1$  and  $\sigma^2$  such that the noise to signal ratio,  $I/C$  is given by

$$I/C = (4\sigma_a^2 / \pi)^{1/2} / |c_1| \quad \text{or} \quad \frac{C_1^2}{\sigma^2} = \frac{4}{\pi(I/C)^2}$$

The  $I/C$  noise to signal ratio determines the length of the Henderson trend-cycle filter to be applied.

Setting  $t=0$  and  $m=6$  for the end weights of the 13-term Henderson, we have

$$E[r_0^{(i,6)}]^2 \frac{1}{\sigma_a^2} = \frac{4}{\pi(I/C)^2} \left( \sum_{j=-i}^6 h_{ij} \right)^2 + \sum_{j=-6}^6 (h_{6,j} - h_{ij})^2$$

Making  $I/C = 3.5$  (the most noisy situation where the 13-term Henderson is applied). The end weights for the remaining monthly Henderson filters are calculated using  $I/C = .99$  for the 9-term filter and  $I/C = 7$  for the 23-term filter. The estimated final trend-cycle is obtained by cascade filtering that result from the convolution of various linear trend and seasonal filters. In fact if the output from the filtering operation  $H$  is the input to the filtering operation  $Q$ , the coefficients of the cascade filter  $C$  result from the convolution of  $H * Q$ . For symmetric filters  $H*Q = Q*H$  but this is not valid for asymmetric filters. Assuming an input series  $x_t, t=1,2,\dots,T$ , we can define a matrix  $H=[h_{kj}]$ ,  $k=1,2,\dots,m_k, j=1,2,\dots,2m_k+1$ , where each row is a filter and  $m_k$  is the half length of the symmetric filter.  $h_1$  denotes an asymmetric filter where the first  $m_h$  coefficient are zeroes and  $h_{m_k+1}$  denotes the symmetric filter. Given data up to time  $T$ , the  $m_{k+1}$  most recent values of the output (filter series) are given by

$$y_{T+1-k}^h = \sum_{j=m_k-k+2}^{2m_k+1} h_{kj} X_{T-k+m_k+2-j}$$

The 13-term Henderson filter can be put into a matrix form:

	N-12	N-11	N-10	N-9	N-8	N-7	N-6	N-5	N-4	N-3	N-2	N-1	N
sym	-0.01935	-0.02786	0.00000	0.06549	0.14736	0.21434	0.24006	0.21434	0.14736	0.06549	0.00000	-0.02786	-0.01935
asym6	0.00000	-0.01643	-0.02577	0.00127	0.06594	0.14698	0.21314	0.23803	0.21149	0.14368	0.06099	-0.00532	-0.03401
asym5	0.00000	0.00000	-0.01099	-0.02204	0.00330	0.06626	0.14559	0.21004	0.23324	0.20498	0.13547	0.05108	-0.01694
asym4	0.00000	0.00000	0.00000	-0.00813	-0.02019	0.00413	0.06608	0.14441	0.20784	0.23002	0.20076	0.13024	0.04483
asym3	0.00000	0.00000	0.00000	0.00000	-0.01603	-0.02487	0.00267	0.06784	0.14939	0.21605	0.24144	0.21540	0.14810
asym2	0.00000	0.00000	0.00000	0.00000	0.00000	-0.04271	-0.03863	0.00182	0.07990	0.17436	0.25392	0.29223	0.27910
asym1	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	-0.09186	-0.05811	0.01202	0.11977	0.24390	0.35315	0.42113

Figure 24 the weights of 13-term Henderson filter  $I/C$  ratio of 3.5

## 7.4 Double Exponential Smoothing for Temperature Extrapolation

In statistics, single exponential smoothing is a technique that can be applied to time series data, either to produce smoothed data for presentation, or to make forecasts. The time series data themselves are a sequence of observations. The observed phenomenon may be essentially random process, or it may be an orderly, but noisy, process.

Whereas in the simple moving average the past observations are weighted equally, exponential smoothing assign exponential decreasing weights as the observation get older. Single exponential smoothing is commonly applied to financial market and economic data, but it can be used with and discrete set of repeated measurements. The raw data sequence is often represented by  $\{X_t\}$ , and the output of the exponential smoothing algorithm is commonly written as  $\{S_t\}$  which may be regarded as our best estimate of what the next value of  $x$  will be. When the sequence of observations begins at time  $t=0$ , the simplest form of exponential smoothing is given by the formulas.

$$S_0 = X_0$$

$$S_t = \partial X_{t-1} + (1 - \partial)S_{t-1}$$

Where  $\partial$  is the smoothing factor, and  $0 < \partial < 1$ . In other words, the smoothed statistic  $S_t$  is a simple weighted average of the previous observation  $X_{t-1}$  and the previous smoothed statistic  $S_{t-1}$ . Single exponential smoothing is easily applied, and it produces a smoothed statistic as soon as two observations are available.

Values of  $\partial$  close to one have less of a smoothing effect and give greater weight to recent changes in the data, while values of  $\partial$  closer to zero have a greater smoothing effect and are less responsive to recent changes. There is no formally correct procedure for choosing  $\partial$ . Sometimes the statistician's judgement is used to choose an appropriate factor. Alternatively, a statistical technique may be used to optimize the value of  $\partial$ . For example, the method of least squares might be used to determine the value of  $\partial$  for which the sum of the quantities  $(S_{n-1} - X_{n-1})^2$  is minimized.

The drawback of single exponential smoothing is that single smoothing does not excel in following data when there is a trend. This situation can be improved by the introduction of a second equation with a second constant,  $\gamma$ , which must be chosen in conjunction with  $\partial$ .

Double exponential smoothing –also known as Holt exponential smoothing – is a refinement of the popular simple exponential smoothing model but adds another component which takes into account any trend over time, simple exponential smoothing forecasts tend to lag behind observation behind observations. Double exponential smoothing is designed to address this type of data series by taking into account any trend in the data. As with simple exponential smoothing, in double exponential smoothing models past observations are given exponentially smaller weights as the observations get older. In other words, recent observations are given relatively more weight in forecasting than the older observations.

There are three equations associated with Double Exponential Smoothing:

$$S_t = \partial y_t + (1 - \partial)(S_{t-1} + b_{t-1}) \quad 0 \leq \partial \leq 1$$

$$b_t = \gamma(S_t - S_{t-1}) + (1 - \gamma)b_{t-1} \quad 0 \leq \gamma \leq 1$$

$$F_{t+1} = S_t + b_t$$

$$F_{t+m} = S_t + mb_t$$

Where

- $y_t$  is the observed value at time  $t$



- $S_t$  is the forecast at time  $t$
- $b_t$  is the estimated slope at time  $t$
- $\alpha$  representing alpha is the first smoothing constant used to smooth the observations
- $\gamma$  representing gamma, is the second smoothing constant used to smooth the trend
- $F_{t+1}$  is one period ahead forecast
- $F_{t+m}$  is  $m$ -periods-ahead forecast

To initialize the double exponential smoothing model,  $S_1$  is set to  $y_1$ . Here are three suggestions for  $b_1$ :

$$b_1 = y_2 - y_1$$

$$b_1 = [y_2 - y_1 + y_3 - y_2 + y_4 - y_3] / 3$$

$$b_1 = (y_n - y_1) / (n - 1)$$

The smoothing constants must be a value in the range 0.0 – 1.0. But the best value for the smoothing constants depends on the data series being modeled.

In general, the speed at which the older responses are dampened is a function of the value of the smoothing constant. When this smoothing constant is close to 1.0, dampening is quick – more weight is given to recent observations – and when it close to 0.0, dampening is slow – and relatively less weight is given to recent observations.

The best value for the smoothing constant is the one that results in the smallest mean of the squared errors.

## 7.5 Algorithms Design for Temperature Extrapolation

The general purpose to design a temperature extrapolation program is to demonstrate the actual estimated temperature behavior in one chart using two algorithms. It is easy to see the difference of two estimated curves comparing with original temperature curve. Four specific days in four seasons are chosen for demonstration because the temperature in different season has different range. The intention is that temperature behavior under different circumstances can be displayed. For example, normally the temperature varies rapidly in summer, so that the curve could be very rough but in winter the variance of temperature at noon and at midnight is not so large, so the curve is shown smoothly. What estimated temperature using two algorithms could be in two situations? Next question is that if it is possible to improve the accuracy of estimated temperature behavior by inserting more observation data points?

Two questions will be answered by means of the programs of temperature extrapolation.

## 7.5.1 Temperature Extrapolation Class Diagram

There are two class diagrams (Figure 24) which extend Application Frame. Application Frame is a base class for creating the main frame for simple applications. Both classes have the attributes which are the standard attributes of JDBC such as statement, connection, results set etc. The array of double “temphourtemperature” contains 24 mean value temperature per hour in one specific day and the array of double “tempyesttemperature” contains the 7 mean value temperature per hour the day before that specific day. The array of double “HendersonEstimatedTemp” contains all the estimated temperature data by using algorithm of Henderson 13 term trend filter and “DoubleExpoSmooEstimateTemp” has all the estimated temperature by using double exponential smoothing algorithms.

The class “TemperatureExtra” has some methods like following:

- **SelectTemperatureFromDB(Calendar oneday)**  
This method takes an instance “oneday” of Calendar class, retrieve temperature data from database according to the value of “one day”. Then the program calculates the mean temperature per hour, obtain the estimated temperature by using two algorithms (Henderson 13 term trend filter and double exponential smoothing) , calculate the sigma between original temperature and two different estimated temperatures. The last step is to plot those data in a xyplot with the help of JFreeChart
- “AverageTemperatureUsingHendersonFilterPerHour” method takes the original “Average temperature”, “EstimatedAverageTemperature” and “YestAverageTemperature” as parameters. The method uses 7 original data as observation points multiplied asymmetric corresponding weights of the 13 term Henderson trend filter to extrapolate three temperatures at the end of time series. Afterward the observation point’s right shift by three points to extrapolate three new temperatures at the end of time series until one day is over.
- “DoubleExponentialSmoothing” method takes the original “Average temperature” double array, “alpha”, “gamma” and “EstimatedAverageTemperature” double array as parameters. This method applies the double exponential smoothing equations and adjust the “alpha” and “gamma” parameter to improve the accuracy of estimated temperature
- “createDataset” and “createChart” methods add estimated temperatures into a dataset and create a xyplot chart to display original temperature curve, two estimated temperature curve.
- “calinvariance” method calculates the mean and sigma between estimated temperatures and original temperatures

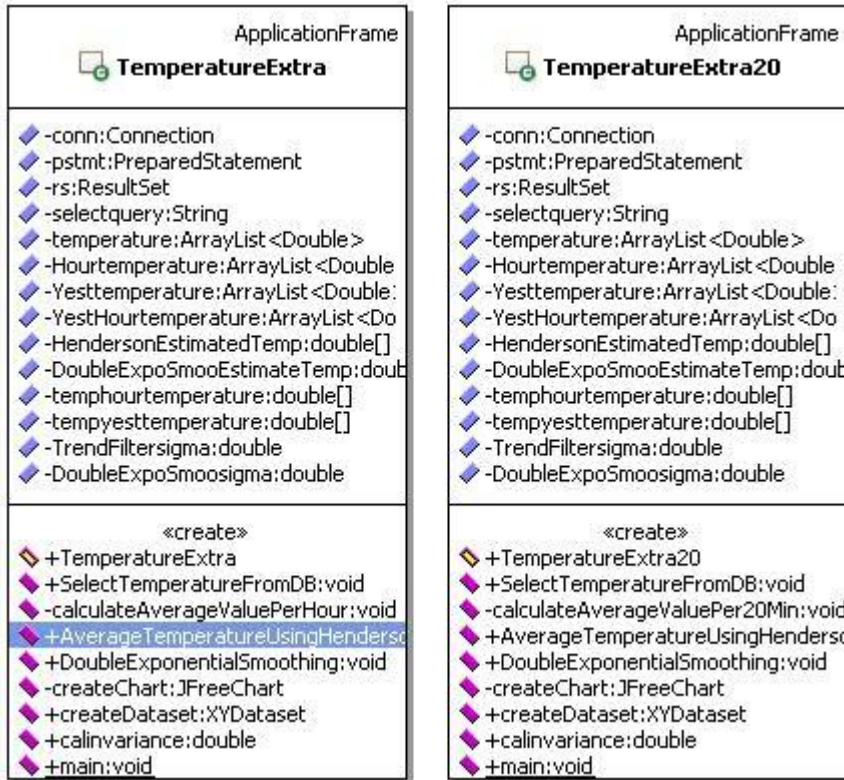


Figure 25 class diagram of Temperature Extrapolation

### 7.5.2 Henderson 13-term Trend Filter Design

Figure 25 is the sequence diagram of “AverageTemperatureUsingHendersonFilterPer-Hour\_13term”. The parameter “EstimateAverageTemperature” is the output which has the estimated temperature. The following asymmetric weights for 13-term Henderson would be used:

(-0.09186,-0.05811, 0.01202, 0.11977, 0.24390, 0.35315, 0.42113)

These weights should be multiplied by the seven original Average Temperatures which are regarded as observation points respectively. One estimated temperature is the result which the multiplication results are accumulated. Next estimated Temperature would be got by six observation points (right shift by one) multiply corresponding weights accumulate the previous estimated temperature multiplied last weight. Next estimated Temperature would be five observation points (again right shift by one) accumulates two previous estimated Temperatures multiply last two weights, so that three estimated temperature will be obtained. All the estimated temperatures can be obtained by this way inside a loop.

“YestAverageTemperature” is used because the data point from 0 to 7 has no old observation points. Seven temperatures at the day before that specific day for the estimated temperature data point from 0 to 7.

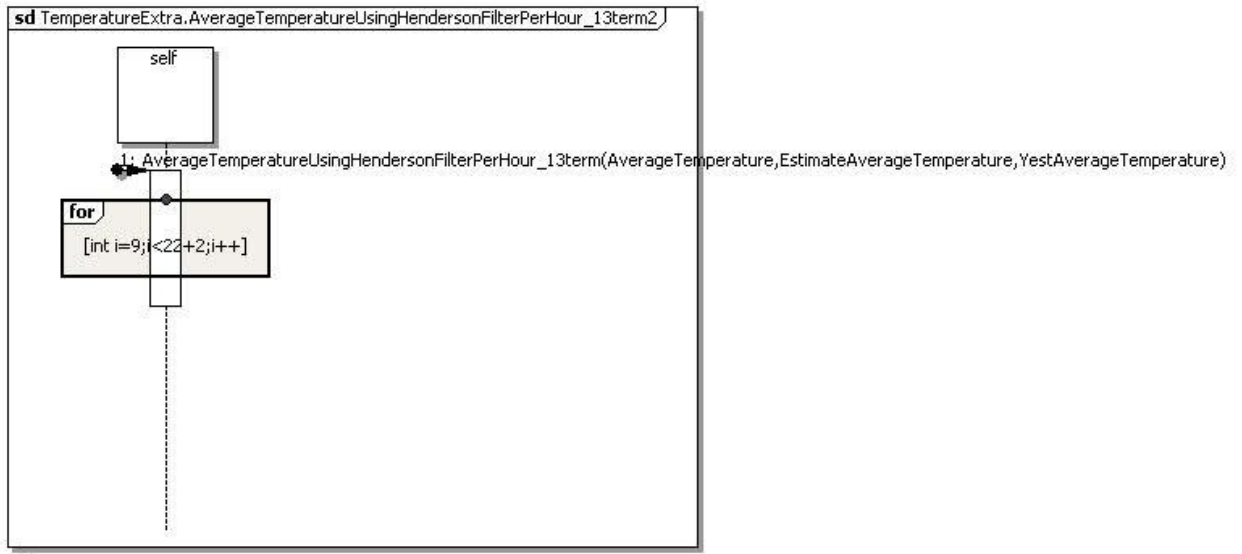


Figure 26 sequence diagram of algorithm of Henderson 13-term Trend filter

### 7.5.3 Double Exponential Smoothing Design

Figure 26 is the sequence diagram of Double Exponential Smoothing. The method “DoubleExponentialSmoothing” applies the following equations into a program.

$$S_t = \partial y_t + (1 - \partial)(S_{t-1} + b_{t-1}) \quad 0 \leq \partial \leq 1$$

$$b_t = \gamma(S_t - S_{t-1}) + (1 - \gamma)b_{t-1} \quad 0 \leq \gamma \leq 1$$

$$F_{t+1} = S_t + b_t$$

$S_t$  is the smoothed temperature,  $y_t$  is the original average temperature,  $b_{t-1}$  is the slope of the temperature trend,  $F_{t+1}$  is one period ahead forecast temperature.  $b_1$  is equal to  $y_2 - y_1$ . The method “DoubleExpoSmoothing” takes a double array “AverageTemperature”, alpha, gamma and a double array “EstimatedAverageTemperature” as parameters. “smoothingAverageTemperature” is temporary double array which indicate the  $S_t$ . “slope” is a array of double which means  $b_t$ . The first “for” loop in figure 26 evaluates the “smoothingAverageTemperature” and “slope” array according to first two equation above. The next for loop evaluate the “EstimatedAverageTemperature” double array which is the sum of “smoothingAverageTemperature” and “Slope” according to the last equation above.

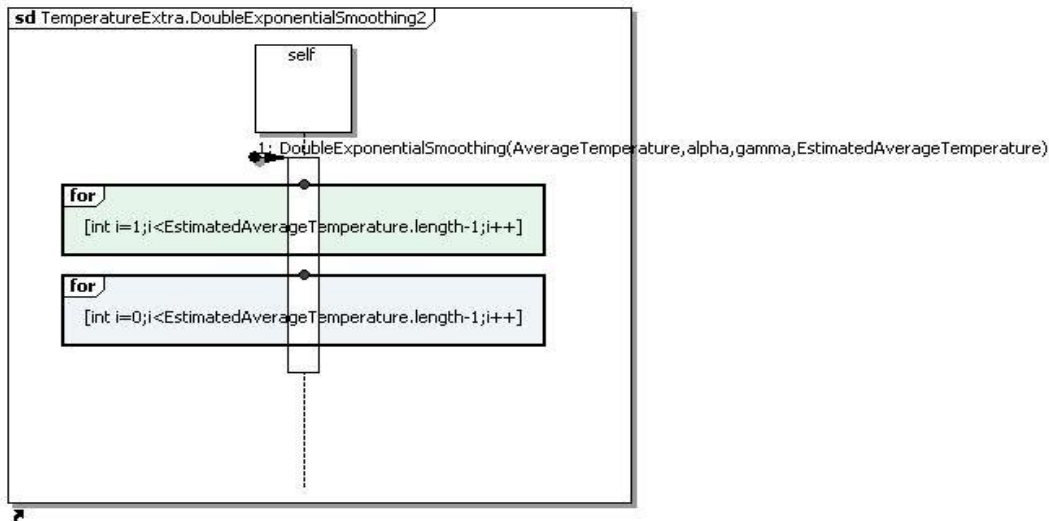


Figure 27 sequence diagram of Double Exponential Smoothing

## 7.6 Temperature Extrapolation Results

The results of temperature extrapolation are shown in the form of xyplot charts. Three days in summer, winter and spring. Two algorithms will be plotted on the same chart plus four curves which indicate the difference between the original temperature behavior and estimated temperature behavior using Henderson trend filter and double exponential smoothing.

Figure 27 is the temperature extrapolation per hour chart on 20-Feb-2009.

er

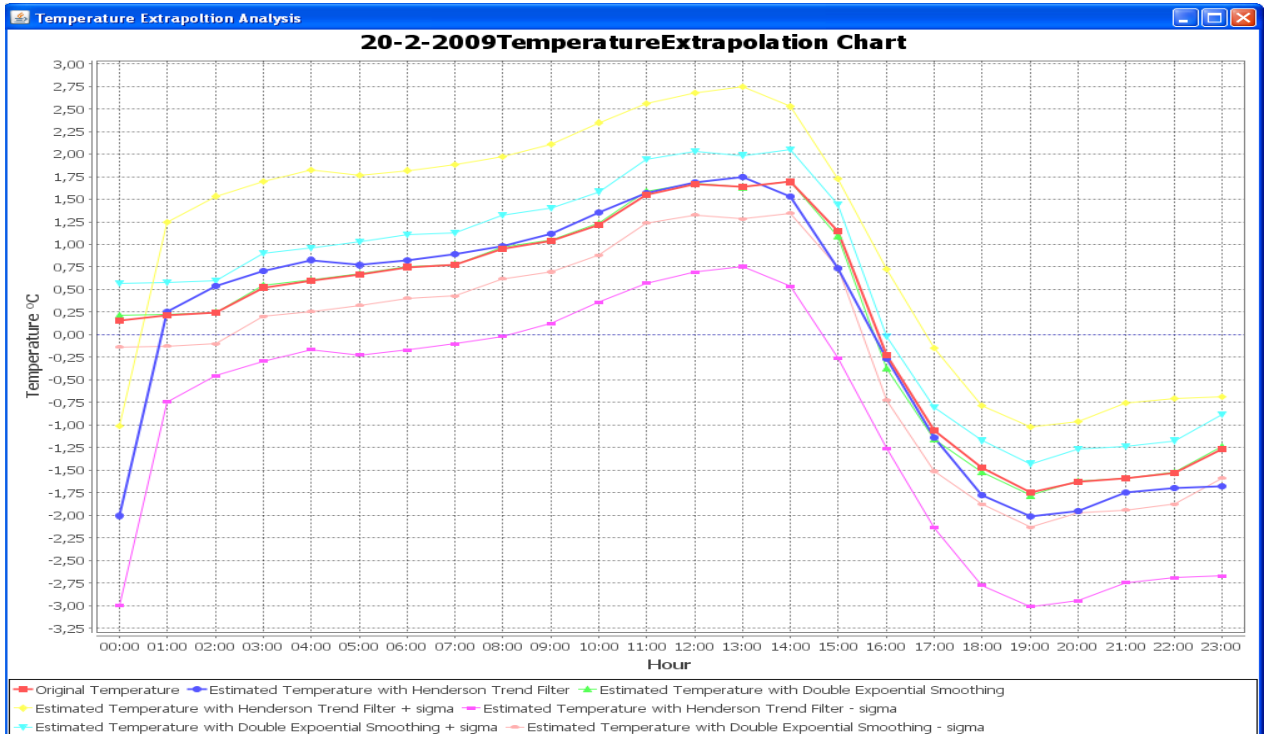


Figure 28 Temperature extrapolation per hour on 20-Feb-2009

That day is a typical day in winter. The temperature is normally below zero degree. It is clearly this temperature curve (red curve) is a relatively smooth curve. The blue curve is the estimated temperature with Henderson trend filter. The mean and sigma value between the original temperature curve and estimated temperature curve with Henderson trend filter is 0.153 and 0.996. The green curve is estimated temperature with double exponential smoothing. The mean and sigma for green curve is -0.054 and 0.5928.

Figure 28 is temperature extrapolation chart per hour on 20-July-2009. It is a typical day in summer. The temperature oscillates in a very short period in one day.

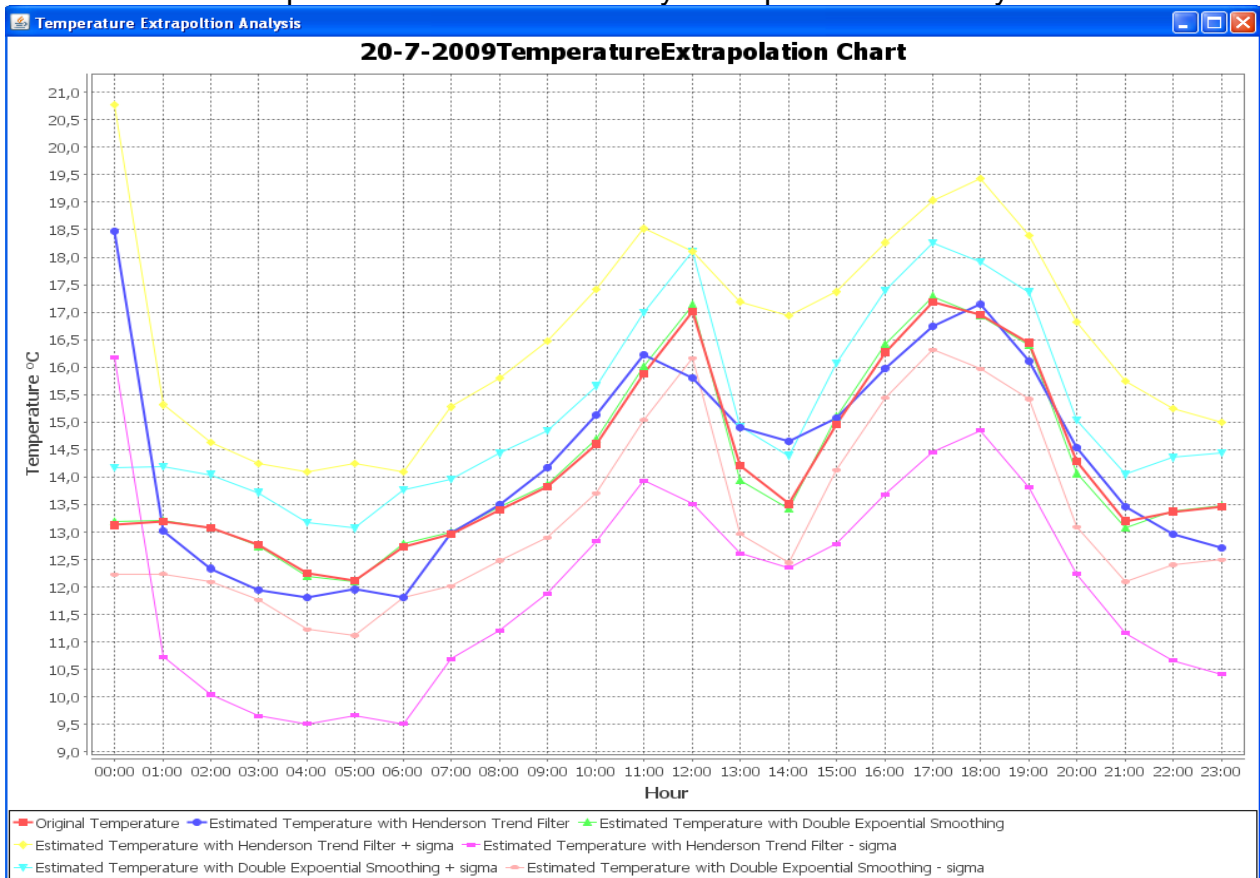


Figure 29 temperature extrapolation per hour on 20-July-2009

In this case, the temperature curve is no longer smooth. The mean and sigma of the error between original curve and estimated temperature curves are bigger. The mean and sigma are -0.579 and 1.513 for Henderson trend filter and 0.009 and 0.987 for double exponential smoothing.

Figure 29 gives the answer if it is possible to improve the accuracy of estimated temperature by adding more data points. Figure 29 is the temperature extrapolation chart per 20 minutes on 20-Feb-2009. The comparison between per hour chart and per 20 minutes chart could be easily made because the same day of temperature is chosen.

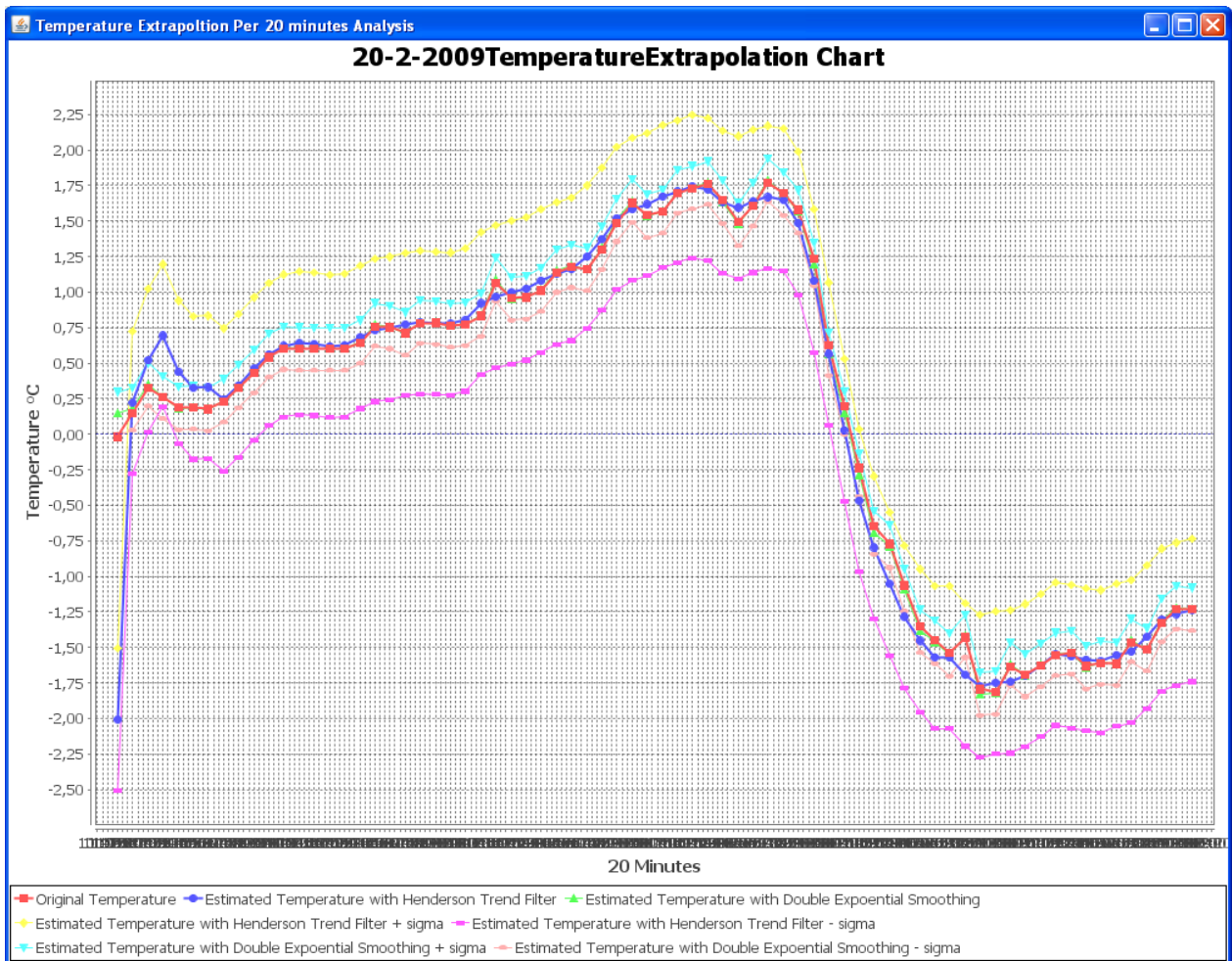


Figure 30 temperature extrapolation per 20 minutes on 20-Feb-2009

The mean and sigma are in this case 0.0458 and 0.708 for Henderson Trend filter algorithm and -0.175 and 0.388 for double exponential smoothing algorithms. The accuracy of temperature extrapolation is really improved by means of inserting more data points. Another proof for improvement of temperature extrapolation is figure 30. In this case, the day temperature and corresponding estimated temperature on 20-July-2009 is depicted. The mean and sigma are -0.174 and 1.1888 for Henderson Trend filter and 0.00378 and 0.8627 for Double Exponential Smoothing. Comparing with the result before, the estimated temperatures for both algorithms are more accurate.

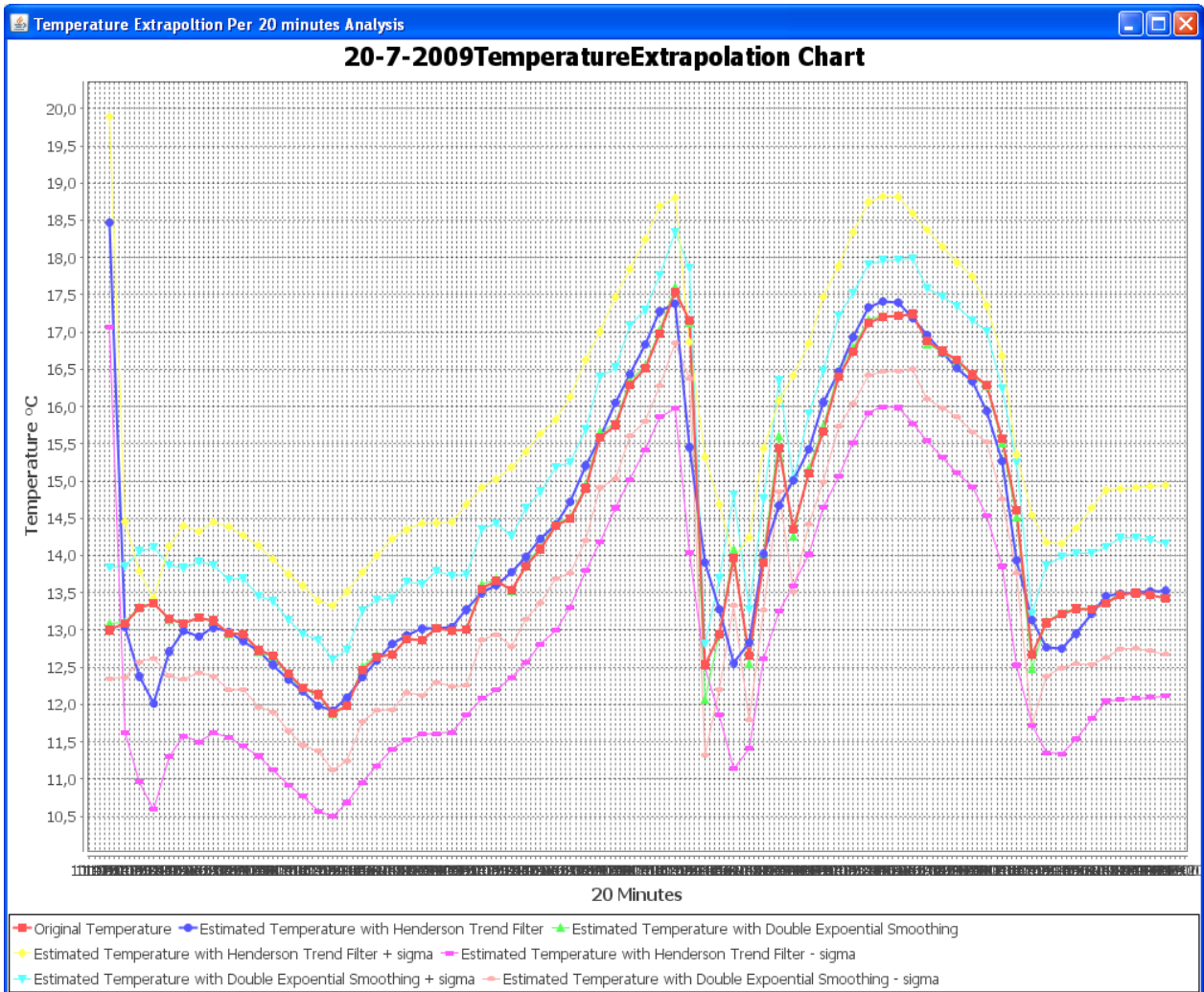


Figure 31 temperature extrapolation per 20 minutes on 20-July-2009

In general case, Double Exponential Smoothing is more suitable algorithm than Henderson Trend Filter in case of day temperature extrapolation. But the problem is that the double exponential smoothing in this case only predicates one point ahead of known temperature. In case one needs to find out what could happen if more temperatures needs to be predicated not only in one day maybe in a week. The algorithms Henderson Trend filter could be better solution because it finds out what data trend become in a relative long period in the future.

In additional, if one wants to apply two algorithms to other weather parameters like pressure, hygrometers, and wind speed and so on. It is also possible as long as the data has a trend.



## 8 Major challenges encountered

Major challenges encountered in the bachelor project are how to construct a nice weather station website and find out appropriate algorithms for temperature extrapolation.

All the GUIs exist on the weather station is based on applets which mean the users' browser must install the java virtual machine, otherwise nothing will work. Another potential problem could be internet connection speed because applets need to be downloaded into users' computer before they are shown on the website. Low speed internet connection probably takes much longer time to download the applets.

Java Applet security policy is another main challenge during this project. The original idea is to select data from database and display it on a website, but it won't be possible because it seriously violate the security policy of applet. Applet is client side application on a web page. If one can select data from database from website, it is also possible to delete table, records even database from a website. That will be a disaster! Applet should never provide methods like database manipulation, file IO and FTP connection. It is too dangerous for the server to be hacked.

Algorithms of temperature Extrapolation could be improved by other algorithms such as linear regression. If another weather parameters need to be extrapolated in a certain time, Algorithms of Henderson Trend filter and Double exponential smoothing should also be adjusted and test.

suggestions for further improvement will be

- Install high JDK version (current version is 1.4) on the server
- Apache Tomcat web server supported in order to handle the connection between applet and database via Servlet ,not file system
- Install a another ODBC-JDBC driver database instead jconnect driver database
- Apply Henderson trend filter and double exponential smoothing to other measured parameters and in a long period.
- Develop a program to find out how to generate the data files when the server suspends for several days. The solution in this case will ignore the date which the data files has not been generated on the server.

## 9 Conclusions

Weather station is important to human's normal life. By presenting the current weather condition of a given place on the web this provides a near to real time update of the weather pattern to the user. Once the data is available on the web, people like to view it in different form. Graphic might be a good way to view the changing characteristic of weather. People browser the weather station website and want to have a general idea about current weather condition and the weather condition in a day, month, or year. They are provided in the weather station website. Three-tier architecture is basic architecture using in the design of the weather station website. JAVA APPLET is main container to present the dynamic web content. File system is bridge to communicate the database and applet.

Weather station website has been constructed and completed in the end. It is located on the webpage of the department information and electrical engineering of the Hamburg University of Applied Science.

Scientist work about temperature extrapolation is the second part of this thesis. Henderson Trend filter and double exponential smoothing are used to predicate temperature in one day. Two simple algorithms have been developed and implemented. With the help of JFreeChart, the results of temperature extrapolation for two algorithms could be seen in a graphical manner. The mean and sigma of two algorithms have been calculated and compared. In the end, the result which algorithm is more suitable for the temperature extrapolation is still negotiable. Two algorithms have their own pros and cons. The future development of these two algorithms is to extend two algorithms to predicate long period temperature and other measured parameters to see if other measured parameters are suitable for two algorithms.

## 10 References

- [1] [http://www.reinhardt-testsys-tem.de/english/climate\\_sensors/weatherstations/weather\\_station\\_MWS\\_6.php](http://www.reinhardt-testsys-tem.de/english/climate_sensors/weatherstations/weather_station_MWS_6.php)  
(MWS 6 weather station) 01.10.2009
- [2] [http://en.wikipedia.org/wiki/Three-tier\\_%28computing%29#Three-tier\\_architecture](http://en.wikipedia.org/wiki/Three-tier_%28computing%29#Three-tier_architecture)  
(Three-tier architecture) 15.10.2009
- [3] <http://en.wikipedia.org/wiki/Typo3>  
(TYPO3) 02.10.2009
- [4] [http://www.ianywhere.com/developer/product\\_manuals/sqlanywhere/1000/en/pdf/db\\_pgen10.pdf](http://www.ianywhere.com/developer/product_manuals/sqlanywhere/1000/en/pdf/db_pgen10.pdf) (SQL Anywhere Server Programming)  
06.11.2009
- [5] [http://en.wikipedia.org/wiki/Java\\_applet](http://en.wikipedia.org/wiki/Java_applet)  
(JAVA Applet) 15.12.2009
- [6] <http://www.jfree.org/jfreechart/>  
(JFreeChart) 07.10.2009
- [7] <http://java.sun.com/javase/6/docs/api/>  
(Java se6 API) 09.10.2009
- [8] <http://www.java2s.com/Code/Java/Chart/CatalogChart.htm>  
(JFreeChart demos) 20.11.2009
- [9] <http://www.weatherforecastmap.com/>  
(Weather forecasting module) 01.12.2009
- [10] <http://www.uow.edu.au/~craigmc/henderson.html>  
(Henderson Trend Filter Description) 04.12.2009
- [11] [http://amsacta.cib.unibo.it/2278/1/quaderni\\_ricerche\\_EBD\\_AC\\_short\\_term.pdf](http://amsacta.cib.unibo.it/2278/1/quaderni_ricerche_EBD_AC_short_term.pdf)

- (Short term trend estimation and turning point prediction)  
18.12.2009
- [12] <http://en.wikipedia.org/wiki/Extrapolation>  
(Extrapolation) 01.01.2010
- [13] <http://en.wikipedia.org/wiki/Forecasting>  
(Forecasting) 09.12.2009
- [14] <http://www.informit.com/articles/article.aspx?p=26316&seqNum=7>  
(Make an applet write a file on the server) 04.11.2009
- [15] [http://ieweb.etch.haw-hamburg.de/yaml3/typo3/alt\\_main.php](http://ieweb.etch.haw-hamburg.de/yaml3/typo3/alt_main.php)  
(TYPO3 inline User Manual) 14.10.2009
- [16] <http://www.itl.nist.gov/div898/handbook/pmc/section4/pmc433.htm>  
(Double Exponential Smoothing) 01.01.2010
- [17] <http://www.itl.nist.gov/div898/handbook/pmc/section4/pmc434.htm>  
(Forecasting with Double Exponential Smoothing LASP)  
05.01.2010
- [18] Head First Java 2nd Edition by Kathy Sierra and Bert Bates 11,May,2009
- [19] Core Java, Vol.2: Advance Features, 8<sup>th</sup> Edition by Cay S.Horstmann and Gery Cornell 18.Apr.2008
- [20] Design of Moving-Average Trend Filters using Fidelity, Smoothness and Minimum revisions Criteria, Alistair Gray and Peter Thomson, Institute of Statistics and Operation Research Victoria University of Wellington

## **11 Appendix**

The appendix contains all source codes, the PDF version of the Bachelor Thesis, all the figures. Additional jfreechart1.0.13.jar, jcommon1.0.16 jar and the jconn2.jar are included in Appendix. All files are on a CD.

This Appendix is deposited with Prof. Dr. Hans-Jürgen Hotop.

### Declaration

I declare within the meaning of section 25(4) of the Examination and Study Regulations of the International Degree Course Information Engineering that: this Bachelor Thesis has been completed by myself independently without outside help and only the defined sources and study aids were used.

Sections that reflect the thoughts or works of others are made known through the definition of sources.

Hamburg, 24-02-2010 , YuanDong Hu