Master’s Thesis

Integration and Verification of Mobile Meteorological Measurements

By

Sergi Collado and Albert Moreno

Department of Electrical and Information Technology
Faculty of Engineering, LTH, Lund University
SE-221 00 Lund, Sweden

Printed in Sweden
E-Huset, Lund, 2011

Abstract

Since 1901, many studies show that the world has warmed by about 0.75 ºC [1]. To face this climate change, the climate community needs to gather more meteorological information due to the poor spatial resolution in some parts of the world. A proposed solution is to use portable sensors connected to mobile phones, like the developed ones in Mobile Meteorological Measurements thesis [2]. The data series collected by these sensors have to be managed in order to be reliable according to the climate environment. It would be a starting point for any decision making in order to increase the mitigation of the climate change.

For this goal, this thesis presents a set of tools to manage this kind of information. For that reason, a server, which receives data series sent by mobile phones, has been implemented. Furthermore, to handle the information, a database has been developed as well as algorithms to evaluate the quality of the data and its properties.

Results shows that the main weakness is that the system has to be customized depending on the final purposes, like agricultural monitoring, microclimate studies or weather prediction. The presented algorithms and its thresholds can change due to the large variety of possible scenarios. As a consequence, more experiments are required to verify the performance of the system in these situations.

Acknowledgments

This Master's thesis would not exist without the support and guidance of Anders Ardö.

The support provided by Jonas Ardö was very useful to improve our algorithms and have a better meteorological background as well.

We thank Fredrik Persson and Tobias Svahn for improving the Android application constantly and performing a good sensor unit box.

We would also like to thank Juanjo Blond, Wenceslao Cruz and Xavier Mallol for their help in the execution of experiments.

Sergi Collado and Albert Moreno

Table of Contents

1 Introduction **1**

1.1 Problem background ... 1

1.2 Proposed solution ... 3

1.3 The thesis ... 3

2 Information sources **5**

2.1 Mobile sensors ... 5

2.2 Internet and weather stations ... 8

3 System architecture **11**

3.1 The server ... 12

3.2 The database ... 13

3.3 Integration and verification ... 13

3.4 Applications ... 22

4 Experiments **25**

4.1 Experiments with stabilized sensors ... 25

4.2 Comparison with a weather station ... 32

4.3 Interpolated map ... 34

4.4 Inside and outside ... 35

4.5 Speed test ... 39

5 Conclusions and future work **41**

5.1 The sensor ... 41

5.2 Android application ... 42

5.3 Server and database ... 43

5.4 Quality assessment ... 44

5.5 Experiments ... 45

References **47**

A. Example of an XML code **49**

B. Database table specification **53**

C. Formulae **55**

List of figures

Figure 2.1: Complete sensor unit with the box opened 6
Figure 2.2: Example of an XML template.. 6
Figure 3.1: Global system architecture.. 11
Figure 3.2: Speed detector... 15
Figure 3.3: Building detector.. 16
Figure 3.4: Probability outside HTC Desire..................................... 17
Figure 3.5: Probability outside HTC Legend 18
Figure 3.6: Quality assessment view 21
Figure 4.1: Temperature measurements with good conditions 26
Figure 4.2: Relative humidity measurements with good conditions 26
Figure 4.3: Pressure measurements with good conditions 27
Figure 4.4: Sensor 1 road in experiment 1. Data overlayed on Google Maps. © 2011 Google - Map data © 2011 Tele Atlas 28
Figure 4.5: Sensor 2 road in experiment 1. Data overlayed on Google Maps. © 2011 Google - Map data © 2011 Tele Atlas 29
Figure 4.6: Sensor 3 road in experiment 1. Data overlayed on Google Maps. © 2011 Google - Map data © 2011 Tele Atlas 29
Figure 4.7: Sensor 1 road in experiment 2. Data overlayed on Google Maps. © 2011 Google - Map data © 2011 Tele Atlas 31
Figure 4.8: Sensor 2 road in experiment 2. Data overlayed on Google Maps. © 2011 Google - Map data © 2011 Tele Atlas 31
Figure 4.9: Sensor 3 road in experiment 2. Data overlayed on Google Maps. © 2011 Google - Map data © 2011 Tele Atlas 32
Figure 4.10: Comparison map. Data overlayed on Google Maps. © 2011 Google - Map data © 2011 Tele Atlas 33
Figure 4.11: Temperature measurement with three sensors fixed.... 34
Figure 4.12: Temperature interpolated surface. The numbers indicates the position of the sensors. Data overlayed on Google Maps. © 2011 Google - Map data © 2011 Tele Atlas 35
Figure 4.13: Temperature measurement with inside outside static .. 36
Figure 4.14: Temperature measurement inside outside walking...... 37
Figure 4.15: Temperature measurement inside outside bike............ 38
Figure 4.16: Speed testing measurement... 39

List of acronyms

GHG	GreenHouse Gases
UN	United Nations
IPCC	Intergovernmental Panel on Climate Change
WMO	World Meteorological Organization
GPS	Global Positioning System
TCP	Transmission Control Protocol
GUI	Graphical User Interface
XML	eXtendible Markup Language
IMEI	International Mobile Equipment Identity
MAC	Media Acces Control
GNU	Gnu is Not Unix
WWW	World Wide Web
IP	Internet Protocol
API	Application Programming Interface
SU	Sensor Unit

Chapter 1

Introduction

1.1 Problem background

The climate has always varied. The main trouble of climate change is that in the last century the pace of these changes has abnormally accelerated to such an extent that it already affects the planetary life. In seeking the cause of this acceleration, scientists found a direct relationship between global warming and the increase of the GHG emissions, most of them caused by the industrialized societies [3]. Furthermore, the increase in the frequency of many other extreme meteorological events such as flooding, storms or hurricanes is an example of the climate change.

Global warming is a term used to describe the global average temperature increase of the Earth's atmosphere and oceans, since 1850 [4]. The UN section responsible for the analysis of the relevant scientific data (the IPCC) argue that most of these observed temperature increases are produced by anthropogenic GHG concentrations [4]. This is known as the anthropogenic theory and predicts that global warming will continue if the human beings do not drastically reduce the emissions of GHG.

As climate stations in large parts of the world are spread thin, the spatial resolution of global meteorological data is sometimes poor. There is a lack of detailed measurements like temperature, relative humidity or atmospheric pressure in large parts of the world, primarily developing countries, where necessary infrastructure is lacking. For instance, in Africa the weather observing stations are averaging one station per 26.000 km^2 [5], eight times lower than the WMO’s minimum recommended level. The sustainable development could be improved if the destruction and loss of life caused by natural disasters are reduced. One way to achieve this is improving the forecasting, multi-hazard early warning systems and disaster preparedness [6].

Any kind of climate change also involves changes in other variables. Their multiple interactions make that the only way to evaluate these changes is by using computer models, which try to simulate the atmosphere and ocean physics [7].

If an increase in the amount of gathered information related to the meteorology is required [1], one possibility is to densify the network of meteorological stations. It could provide very valuable information in forecasting natural catastrophes like floods or hurricanes in a near future.

Instead of building new weather stations, one solution would be to use mobiles phones to collect meteorological data, since they are found almost everywhere in the world. Moreover, the constant improvement in their technology is a fact. The arrival of Internet connection to them, the data transfer via Bluetooth and the GPS positioning make the mobile phone the best device for this goal. The solution is to add meteorological sensors to the mobile phone as well as verifying tools to develop this data and presenting them so that future decision-making has the greatest accuracy possible.

The main feature of mobile phones as well as the used sensor is their portability. But it is a double-edged sword. On one hand, it offers possibility of having meteorological information in all the places where the user is, but on the other hand, the data will not always be reliable. In some cases the meteorological information that is needed should be taken in the open air, while in other cases it would be useful inside buildings or industrial estates for example.

To sum up, having an extensive network of meteorological sensors could provide complementary data to the current types of measurements to better analyse cases like the gradual desertification of wooded areas, changes in rainfall patterns worldwide, floods, etc. more effectively. It might involve a powerful tool for making decisions to counteract the effect of climate change.

A part from that, other specific applications can be taken into account where parameters of the sensors (location or mobility for instance) could be known, allowing the customization of the system to be designed in a more precise way.

Examples of those applications are:

- Agricultural high resolution monitoring
- Studies related to health
- Studies for urban environments
- Better performance of the current weather prediction models
- Temperature, humidity and pressure monitoring in industrial facilities
- Heat wave or cold spell prediction/monitoring

1.2 Problem specification

This thesis aims at investigating the possibilities of collecting and processing data sent by a mobile phone connected to a portable meteorological sensor. For this goal, the problem deals with the creation of an appropriate server to receive primitive weather information. The creation of a database that could be able to be connected to the server and save time series of meteorological measurements turns out fundamental. Last but not least, the investigation of possible ways or methods to verify these data is required as well. Experiments to verify the performance of the system will be carried out.

1.3 The thesis

In chapter 2 sources of information available for the system are shown.

Chapter 3.1 describes the design and implementation of a server. Chapter 3.2 deals with the creation of a database. Moreover, in chapter 3.3 the integration and verification of the gathered data is presented.

In chapter 4, the system is tested with experiments to check the processed information reliability after being collected by the sensors.

Finally, in chapter 5 conclusions and future work are presented.

Chapter 2

Information sources

In this part, sources or possible sources of meteorological information for the system will be explained.

2.1 Mobile sensors

In order to develop this project, meteorological sensors created by Fredrik Persson and Tobias Svahn (both students of Department of Electrical and Information Technology, Lund University) [2] will be used.

The main source of information is the sensor unit and therefore, the mobile phone, which is connected to the sensor via Bluetooth.

The most important variables taken are temperature [ºC], relative humidity [%] and atmospheric pressure [hPa]. The phone sends to the server the information taken by the sensor through TCP/IP connection for its subsequent processing. If there is no Internet connection at the moment of sending the data, it can be stored in a file in the phone by the user, who will send the file afterwards when there is Internet connection again.

The chosen phones for experiments were an HTC Legend and two HTC Desires, all with Android 2.2 operating system (Froyo). In the developed GUI application for the mobile phone there are four menus:

- Standard Series: gets values every minute.
- Custom Series: allows the user to change the measurement period.
- Test Measurement: makes only one measurement and shows all the information in the screen.

- Settings: this menu is used for scanning sensors. The user can also choose if the information is going to be sent to the server automatically or stored in an internal file.

As it has been said, the main feature of the sensor is their portability since it is 85x40x56mm (LxHxW).

Figure 2.1: Complete sensor unit with the box opened

Every measurement gathered by the sensor is converted to an XML template. The structure of its template is the following:

```
<data>
    <tag1>description_tag1</tag1>
    <tag2>description_tag2</tag2>
              ...
              ...
    <tagN>description_tagN</tagN>
</data>
```

Figure 2.2: Example of an XML template

It is worth noting that it is not necessary that all the variables neither have to be sent nor in order. XML code series can be sent without some variables, since not always all the information could be obtained, in which case it will be stored a NULL value in each non-sent variable. Besides, more than one XML template can be sent in one file.

These are the description of the tags:

- Source: It shows the origin of the information
  - "1" if the information comes from weather stations.
  - "2" if it comes from Internet crawled data.
  - "3" if it comes from mobile sensor units.

- IMEI: International Mobile Equipment Identity. It is a number, usually unique, to identify mobile phones, as well as some satellite phones.

- Day: Day when the information is sent. The format is: 'yyyy-mm-dd'.

- Time: Time when the information is sent. The format is: 'hh:mm:ss'.

- Position_latitude: GPS latitude from where the mobile phone sends the data. Units: ° (degrees).

- Position_longitude: GPS longitude from where the mobile phone sends the data. Units: ° (degrees).

- Position_altitude: GPS altitude from where the mobile phone sends the data. Units: m (meters).

- Position_accuracy: Identifies the GPS positioning accuracy. Units: m (meters).

- Bmptemp: Temperature obtained by the pressure sensor. Units: °C (degrees Celsius).

- Temperature: Temperature recorded by the temperature sensor. Units: °C (degrees Celsius).

- Humidity: Relative humidity saved by the humidity sensor. Units: %.

- Pressure: Pressure obtained by the pressure sensor. Units: Pa (Pascal).

- Light: Light recorded by the light sensor of the phone. Units: lm (lumen).

- Lightsensor: Name of the mobile phone light sensor.

- Accelsensor: Name of the mobile phone acceleration sensor.

- Accelerationx: Acceleration on the x-axis defined by the phone.

- Accelerationy: Acceleration on the y-axis defined by the phone.

- Accelerationz: Acceleration on the z-axis defined by the phone.

- Comment: Comment added by the user at the moment of measuring the file.

- Sensor_mac: MAC address of the mobile phone.

- Phonemodel: Model of the mobile phone.

In appendix A there is an example of an XML file with three XML codes.

2.2 Internet and weather stations

As well as utilizing the sensors, there is more information that is useful. There are many websites related to meteorology that provide this kind of values like temperature, humidity and pressure. This information could be compared to data gathered by sensors in order to verify these data. It will make sense if the place where the sensor is and the place where the weather station of the web page measures are near enough to conclude that there are the same or similar weather conditions, because the difference in altitude between the sensor and the station can lead into a high difference in the climate conditions.

Thus, the system has to work automatically. There is software that allows doing that. A web crawler [8] is a computer program that browses the WWW in a methodical, automated manner or in an orderly fashion. This process is called web crawling or spidering. Many sites, in particular search engines, use spidering as a means of providing up-to-date data. A web crawler could be used to search meteorological information on Internet for its later comparison with data gathered by the sensor.

Other sources of information are weather stations. To date November 1st, 2010, climatological data from 1322 cities corresponding to 162 WMO members were available. 124 members of these 162 send weather forecasts to 1319 cities as well. These members supply national meteorological or hydrological services in their countries [9].

Moreover, the World Data Center for Meteorology, hosted by the NCDC (National Climatic Data Center) in USA provides records from more than 6,000 stations around the world, which have reported at three-hourly intervals for at least 15 years [10]. Much useful information is accessible and can be gathered for several purposes.

It is worth saying that the system was mainly designed using the sensor and the mobile phone as the main source of information.

Chapter 3

System architecture

As seen in the previous section, the system (Figure 3.3) has a number of inputs to produce some specific applications to the final users. In order to receive the data, a server based on a C language running under GNU/Linux platform has been implemented. The data storage will be managed by a MySQL database. Then, the computational power offered by Matlab will be used to implement algorithms to detect errors and verify both reliability and quality of the information. Once this information is processed, different kind of applications can be made depending on the final user.

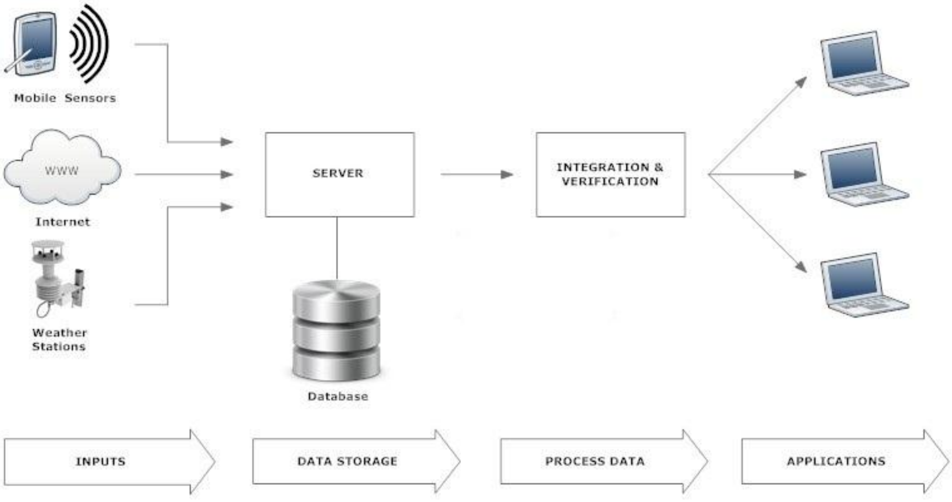

Figure 3.1: Global system architecture

3.1 The server

To establish a connection between inputs and our system, a C based server under GNU/Linux platform has been designed.

Linux was chosen because it is an open source operating system that performs many features and flexibility to develop a server, like:

- Multithreading and multiuser
- Multiplatform and multiprocessor
- Good permissions management
- Memory protection between threads
- Core dumps

Moreover, the C programming language offers a number of tools to establish a connection between many clients and the server, called sockets. The chosen protocol is TCP/IP which guarantees that the data will be received properly.

Finally, the server type is a concurrent server. This kind of server offers good connections management due to the fact that the listening socket can be held by multiple clients. Furthermore, since the load is currently low, the server does not have any kind of preforking, in other words, it creates every child thread when a new connection is requested.

In addition, the server makes a backup of all incoming information, which should be stored in an external server, to avoid possible loss of information at any time.

In order to handle information the system receives, it follows these steps for every connection request:

1) The server stores all the XML templates received in a file

2) Automatically makes a backup that has to be stored in an external server

3) It reads the XML codes and inserts all the information in the MySQL table called *Incoming_Data*

4) Manage data series in order to prepare it before being processed by the quality assessment tools

5) After using the algorithms, it stores the output information in a table called *Real_Data*

3.2 The database

To store all data properly, the server must be connected to a database. Since the amount of data will not be too high for our tests, a MySQL database, which is under an open source condition, has been set up.

MySQL offers the following features:

• Good speed performing operations, using the multiprocessor systems power, due to its multithreaded implementation.
• Low resource consumption.
• Big number of functions available through its API for C.
• Good portability between systems

The database consists of two tables. On the first one all the data sent by the mobile phone is stored (*Incoming_Data* table) and on the second one the information processed by the system is saved (*Real_Data* table). The fields of the table correspond with the tags of the XML template.

See appendix B to find all the fields of these tables.

3.3 Integration and verification

After received data is stored, it should be processed using algorithms. Afterwards, there will be a study of the data reliability and its quality, since information gathered by the sensor may not be realistic according with the real weather conditions. This study will show the probability of being in the required scenario according with the final users purposes. For instance, most applications only need climate information, therefore data gathered by the sensor inside buildings will not be useful and the system has to remove them.

To perform all these functions, the data will be processed in a Matlab environment. It offers a great computationally power, advanced graphics visualization and a very useful collection of great signal processing functions.

3.3.1 Detectors

As it has been said previously, for a better performance of the data, the system has to detect in which kind of situation the user is. To do this, four detectors have been created which should be understood as a tool for analysing incoming information.

1. Length detector

It is responsible for deciding whether an experiment is valid or not depending on its duration. The user can set a maximum duration or minimum for this. If the experiment is outside the set range, the data will be discarded.

2. Gaps detector

As mentioned in section 2.1, synchronization between each mobile and each sensor unit is via Bluetooth. Depending on the Bluetooth version and the distance between the phone and sensor unit, there may be disconnections. When the phone is disconnected from the sensor, it stops taking data and the user must manually restart the Android application to resume sending information.

In this respect, this detector provides useful information on the duration of the gaps in case of disconnection of Bluetooth and can be considered that an experiment with many long-time disconnections is not valid.

Both HTC Desire [11] and HTC Legend [12] have a Bluetooth 2.1 device which can reach a speed of 3 Mbps [13].

3. Speed detector

Meteorological data may vary depending on whether the sensor unit carrier is stationary or moving. To detect this situation, data collected by the GPS antenna from both HTC Desire and HTC Legend will be used (Figure 3.2).

The following algorithm diagram is used:

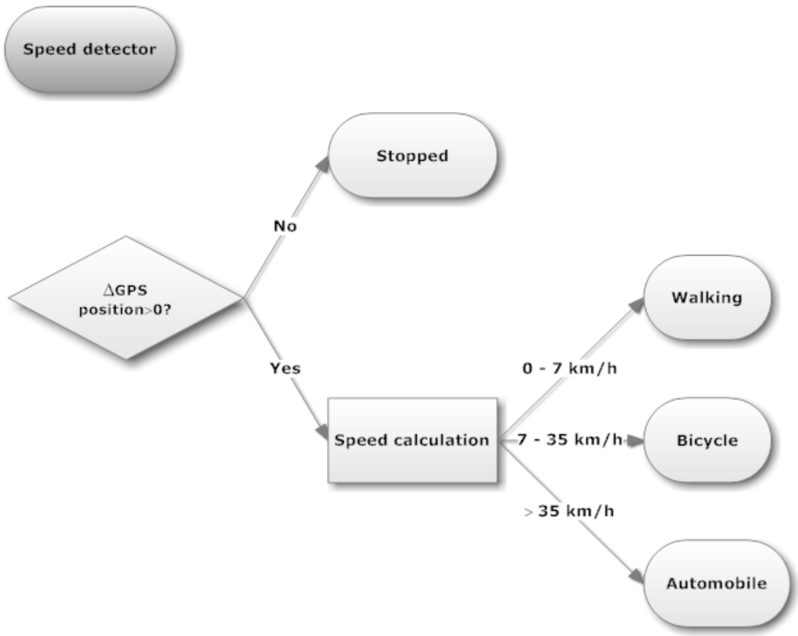

Figure 3.2: Speed detector

In appendix C.1 are the speed calculation equations.

4. Building detector

The study of meteorological information is usually based on samples which are taken outdoors. This detector corrects the possibility that the user takes samples from inside buildings according to the GPS location and then checks that the measurements taken by the sensor are real as follows.

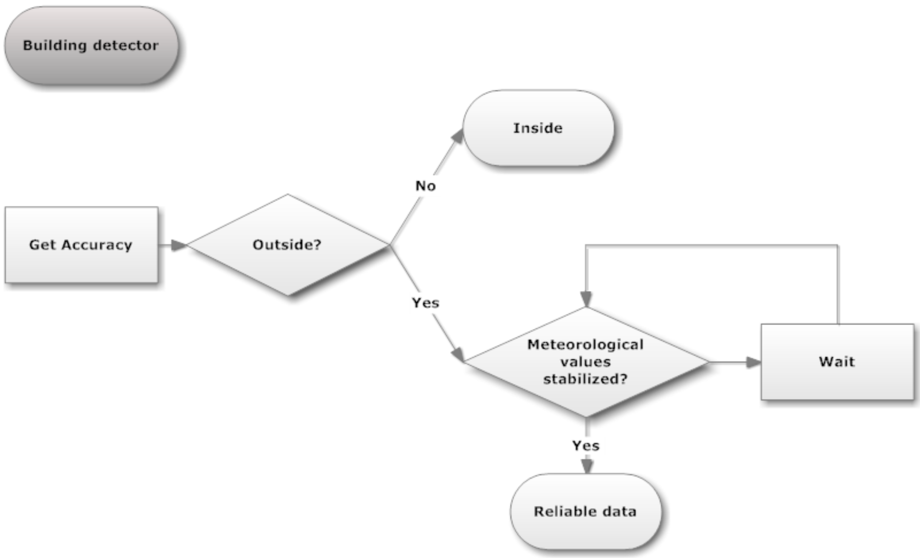

Figure 3.3: Building detector

Step 1: GPS accuracy

In this first step GPS tracking is used. The mobile phone receives signals from at least 3 satellites, calculates the distance to each satellite and then determines the geographical position [14]. This process involves a degree of accuracy, which is based on the quality of each of the signals received by satellites. The accuracy is expressed in meters and is accessible from every mobile phone. Since it is related to the imprecision of the measure, low values of accuracy mean a good geolocation and high values mean a bad geolocation.

According to the manufacturer specifications, the chipset that is responsible for receiving the GPS signal is Qualcomm Snapdragon QSD8250 1 GHz [15] for HTC Desire and Qualcomm MSM 7227 600 MHz [16] for HTC Legend. This chipset is a unit that integrates CPU, GPU, and 3G mobile broadband and a built-in GPS engine with Assisted-GPS, which enhances GPS low power signal reception [17].

Obviously, outdoor situations where sources of interference are low will have a high accuracy. On the other hand, when the user is near or inside buildings, the value of the accuracy will be high, not only because the walls and ceilings make the good reception of satellite signals difficult but also by other factors such as interferences and noise in the radio signal.

To better detect what situation can be found, a study of the probability of being inside a building depending on the accuracy of the GPS signal has been made. For this purpose, it has been written down the accuracy of each mobile phone in different situations such as:

- Inside a building over 10 m from the exterior walls.
- Inside a building near an exterior window.
- Inside a building near an exterior door.
- Outside a building near a gateway.
- Outside a building closer than 15 m from the exterior walls.
- Outside a building over 15 m from the exterior walls.

The building chosen was the E-building, Lund.

The following graphics relate the probability of being outside to the GPS accuracy according to the previous cases.

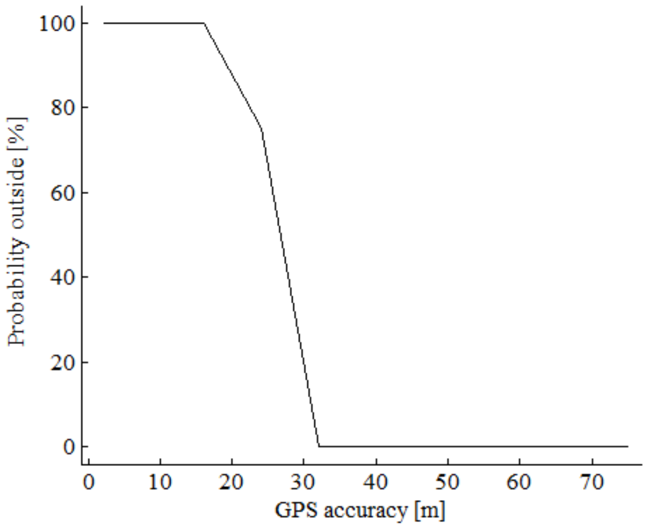

Figure 3.4: Probability outside HTC Desire

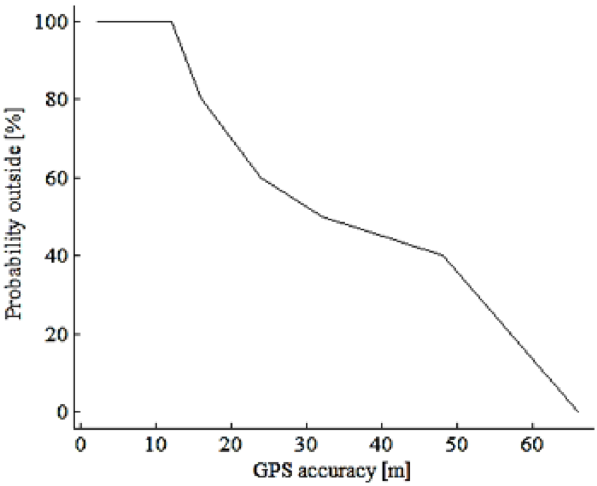

Figure 3.5: Probability outside HTC Legend

It is observed that the HTC Desire has always been outside a building when the GPS accuracy was below 16 m (Figure 3.4). Moreover, values above 32 m indicate that the quality of GPS signals is very low because the phone has always been inside a building. For values within this range, uncertainty decreases rapidly.

For HTC Legend, although there is a 100% of probability to be outside a building for values lower than 12 m, uncertainty is much higher since this probability decreases much more smoothly. To assert that the SU is inside a building accuracy has to reach higher values than 66 m (Figure 3.5).

It is worth mentioning that although for low levels of accuracy the SU has always been outdoors, it is impossible to be a hundred per cent sure that it will happen in all cases. Not only users will not work ever with the same mobile phones, fact that will change the shape of the probability graphic, but there are many situations where the mobile will lose GPS coverage as well. To make these graphics, 400 GPS accuracy samples have been taken. Depending of the accuracy required for the final application, a minimum probability of being outside or inside can be established. After this threshold is set, the detector decides where the sensor unit is.

Step 2: Information reliability study

Once it is detected that the sensor location is outside, not all the gathered information will be reliable according with real meteorological values as sensor inertia introduces an error. When an abrupt temperature change is detected (for instance in winter it can go from 20 °C inside a building to 0 °C outside) the sensor cannot reach the new temperature instantly. This process may vary due to several factors such as ventilation of the sensor, the movement of the person carrying it or the temperature difference between one moment and another. These factors affect the sensor speed to take data close to the new temperature of its environment.

For this reason, the next algorithm has been developed:

1. The information taken by the sensors is compared with reliable meteorological data. If they are similar it continues to the step 2.

2. Here, the inertia of the measure is studied. To do so, it is compared the first derivate of the measure to a threshold. If the derivate is short, it means that the sensor is stabilized and series can be considered like reliable data.

5. Weather station comparison

This algorithm uses a database with a weather station list. For each gathered measurement, it performs the following steps:

1. Compares the measurement GPS position with all the GPS positions of this weather station list.

2. Looks for the nearest weather station. The main problem for this comparison is the difference in altitude between the sensor and the station, which can lead into a high difference in the climate conditions.

3. Compares the meteorological values between this weather station and the sensor values and stores the difference in the *Real_Data* table.

Afterwards, as it is analysed in the experiment 4.2, it can be considered a radius to decide in which situations this data can be useful.

3.3.2 Quality assessment

Once the collected information is processed, a tool to visualize the quality of the system has been developed. This tool shows the following parameters:

- Length of the experiment
- Maximum duration of time without data
- Probability of being inside a building
- Speed [km/h]
- Measurements average
- Measurements variance
- Difference with the closest weather station
- Environmental light
- Mobile phone acceleration

The user has to define thresholds according to the needs. Each factor can be disabled.

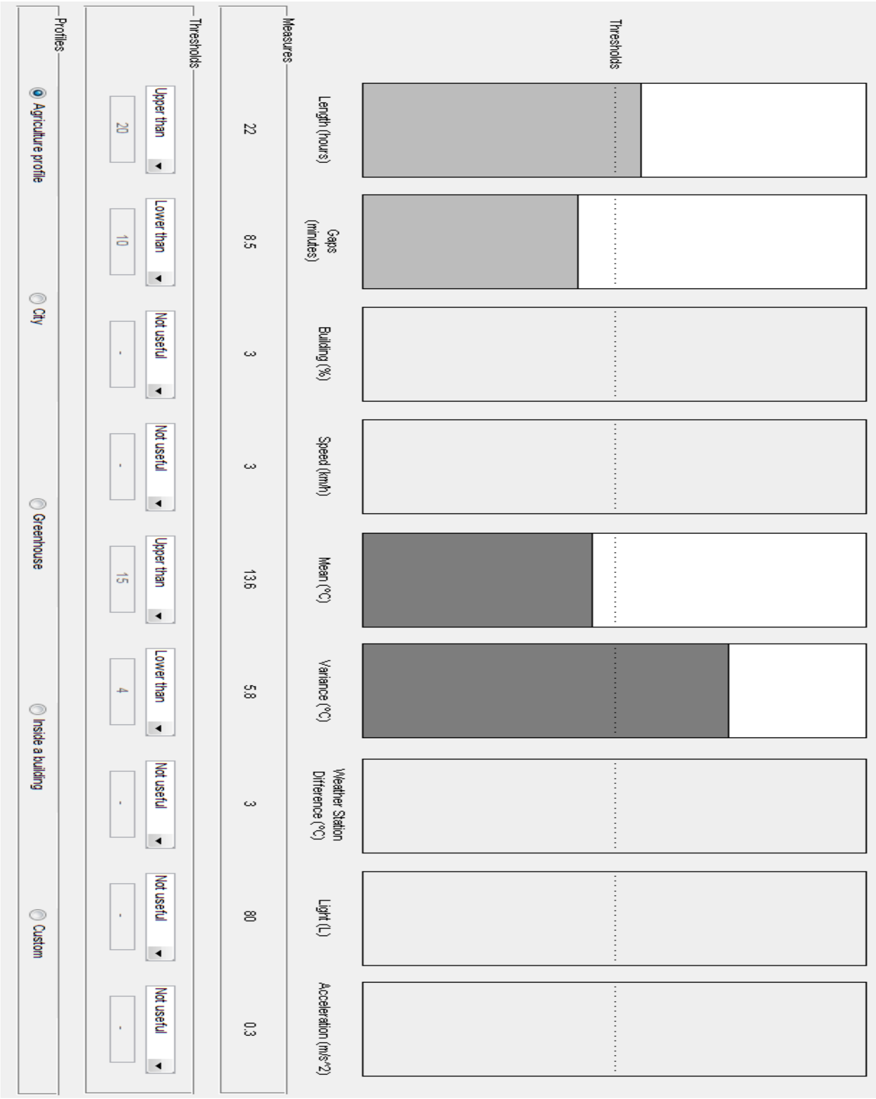

Figure 3.6: Quality assessment view

Besides, in the lower bar there is a profile's button. Each profile has its own parameters depending on the needs of each application. On the one hand for example, if the sensors are placed in an agricultural environment to control the temperature and humidity, it is not necessary to activate the building detector since the sensors are fixed and outside a building. On the other hand, studies of urban climate need information about the geographical position of the sensors. Furthermore, it is allowed to customize all the thresholds manually.

3.4 Applications

So far the project has focused on assessing the quality of information. Once the information is more realistic, the last goal of this thesis is to show useful tools for final users, which can have a diversity of features.

One possibility is to use meteorological data at a specific time and make maps of a geographic area in order to see graphically the information collected by all the sensors in that zone and the nearby professional weather stations as well. In order to make maps the data is converted to a standard track file and processed by GPS Visualizer [18].

There are three ways to see the information:

1. Time maps: Data collected by a sensor is depicted throughout a map.

2. Multi-sensor maps: In this case, information of more than one sensor is shown on the same map. The data are registered at the same time.

3. Interpolated surfaces: It is based on an algorithm, which uses meteorological data from sensors situated in several geographic locations to make an interpolation. This process adds spatial resolution for an instant of time. The result is a temperature values surface for example, which is drawn on a map. The more sensors used per km^2, the higher accuracy of the surface will be obtained.

In chapter 4, there are different maps that have been made from several experiments with three sensors around Lund.

Chapter 4

Experiments

This chapter describe and report the experiments performed for verification of the operation of the sensor and the algorithms implemented.

4.1 Experiments with stabilized sensors

This section introduces the basic experiments that have to be done in optimal conditions. Before making the measurements the sensors were:

1. Placed outside
2. Completely stabilized

4.1.1 Outside

As it was known that the sensor would take a while to stabilize the measurement, this experiment was made with a sensor in optimal conditions, that is to say, the sensor was placed outdoors before the experiment for 10 hours.

The information collected by the sensor at the open air was compared to the data of a weather station in Lund [19] which is 1 km away. The temporary resolution of the sensor is one sample per minute (standard measurement), while the resolution of the weather station is one sample every thirty minutes.

This graphic shows the sensor temperatures and the weather station respectively:

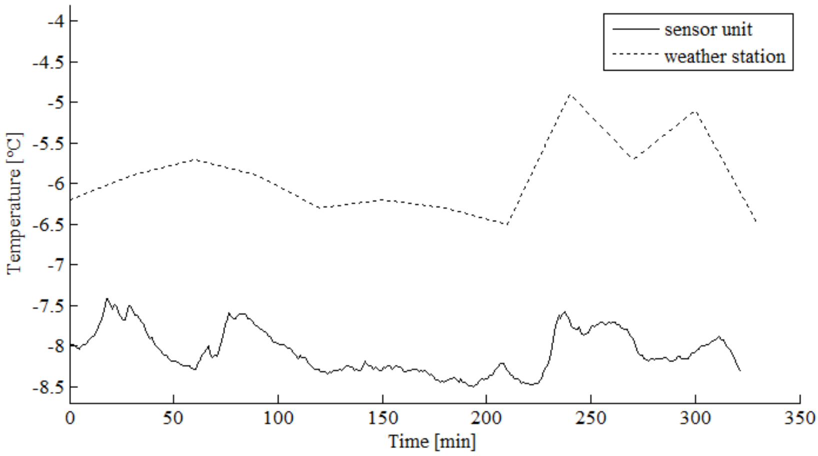

Figure 4.1: Temperature measurements with good conditions

At first sight it is observed that the weather station registered between 1.5 ºC and 2 ºC above the temperature of the sensor during the experiment. The shape of the graphic is quite similar in general, which means that between one source of information and the other there is an approximate constant offset.

The relative humidity data was also analysed:

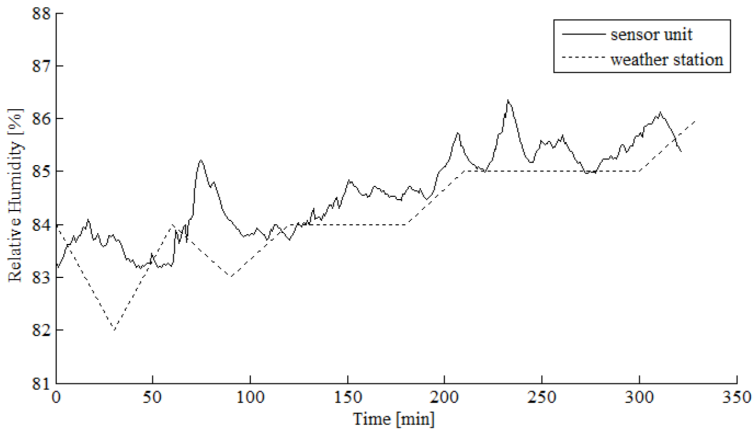

Figure 4.2: Relative humidity measurements with good conditions

The relative humidity of the sensor was significantly higher throughout the experiment. This may be due to the closed form of the box containing the sensors, which does not have optimal ventilation for the collection of samples of humidity. It is also noted that in minutes 30 and 90, the humidity of the weather station dropped, while the sensor rose slightly. Although there is no direct explanation for this result, one possible cause may be an error on the humidity measurement made by the weather station.

Finally, the pressure measurements were studied:

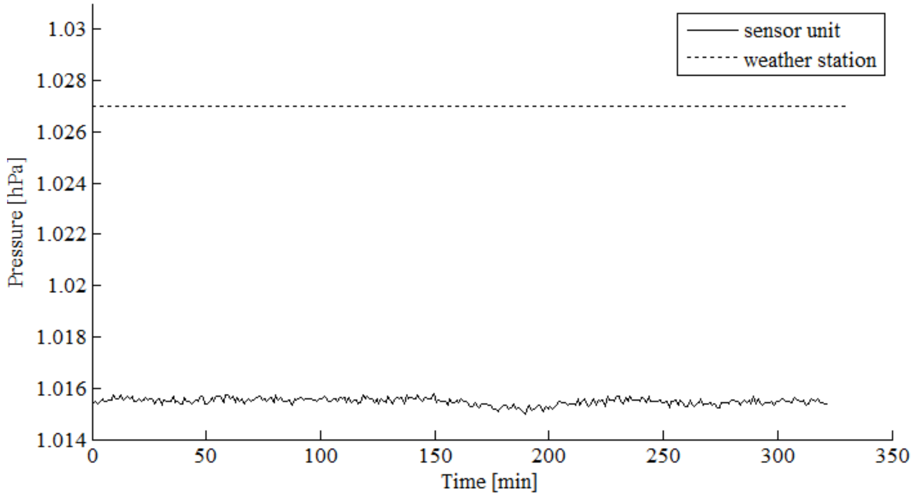

Figure 4.3: Pressure measurements with good conditions

Figure 4.3 shows a clear and constant variation in pressure between the two graphics which is a 0.0115 hPa offset. It may be due to a calibration error on one of the pressure sensors.

4.1.2. Experiments with three sensors

To study in detail the performance of the sensors in an urban environment in Lund (Sweden), two tests have been made with the three available sensors.

<u>Experiment 1</u>

In this case three people participated in the test, each of which carried a sensor hung on the hand and a mobile phone and took three walks. The following roads were covered on January 19th, 2011 from 4 pm to last approximately one hour and forty minutes:

- Sensor 1: from Skarpskyttevägen 226 (point 1) to Mårtenstorget (point 2).
- Sensor 2: from Mårtenstorget (point 1) to Mårtenstorget (point 2) through the west of Klostergården.
- Sensor 3: from Mårtenstorget (point 1) to Mårtenstorget (point 2) through the east of Klostergården.

The three roads are described in a picture. The temperature is represented in a colour scale according to its value.

Figure 4.4: Sensor 1 road in experiment 1. Data overlayed on Google Maps. © 2011 Google - Map data © 2011 Tele Atlas

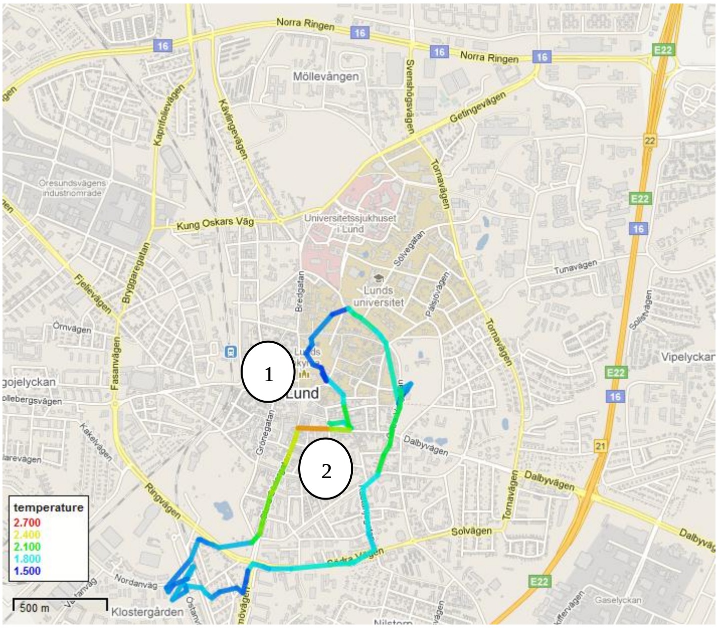

Figure 4.5: Sensor 2 road in experiment 1. Data overlayed on Google Maps. © 2011 Google - Map data © 2011 Tele Atlas

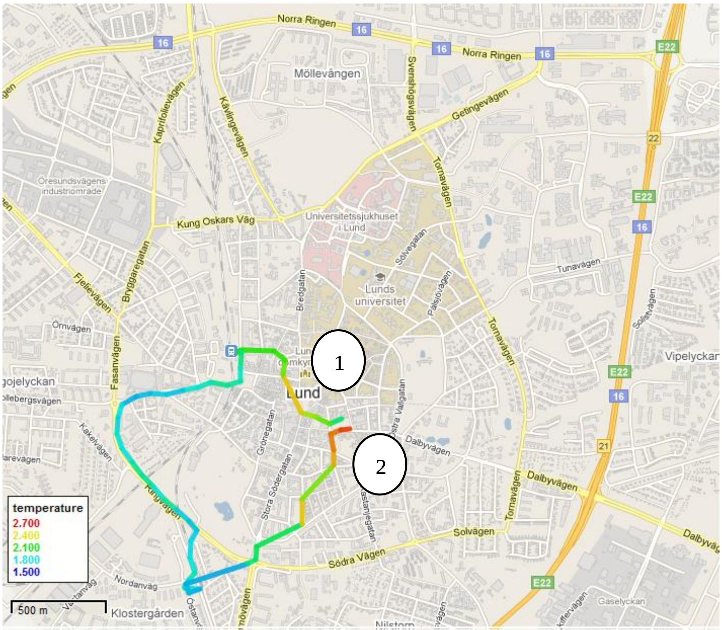

Figure 4.6: Sensor 3 road in experiment 1. Data overlayed on Google Maps. © 2011 Google - Map data © 2011 Tele Atlas

As it can be seen, the maximum difference during the experiment is 1.2 ºC. The sensor 1 is at lower temperature than the rest at the beginning of the experiment. This may be due the placements of the sensors, since number 2 and 3 were on the city centre while number 1 was on the outskirts.

Five minutes later from the start, the sensor 1 registered an increase of the temperature, being constant between 2 ºC y 2.6 ºC until the end of the test. Sensor 2 detected a decrease of the temperature reaching about 1.8 ºC when it was close to the west of the city and an increase at the end close to Mårtenstorget.

The sensor 3 showed a clearer variation of temperature. During most of the road, the temperature ranged from 1.8 ºC to 2.5 ºC but there were three instants where it dropped to 1.8 ºC, which corresponded with the moments where the sensor was in Klostergården, Botaniska trädgården and Lundagård. Although the difference is lower than 1 ºC it may be due the fact that they are either parks or far from the city centre at a lower temperature.

Finally, it is considered that the experiment showed fairly consistent results with reality, since the weather station located in the E-Huset (near John Ericssons Väg) reported values between 1.1 ºC and 1.3 ºC during that period of time. The degrees of difference between the station and the sensors could be due to ventilation, as the station has got a better one than the ventilation of the sensors.

Experiment 2

This experiment is similar to the previous one. The influence of covering the ventilation holes of the sensor 3 in comparison with the rest was studied. The sensors 1 and 2 were hung on the hand. Here the three followed roads started and ended on Mårtenstorget. The sensor 1 covered the north part of Lund, the sensor 2 covered the east zone and the sensor 3 the southwest part of the city.

Figure 4.7: Sensor 1 road in experiment 2. Data overlayed on Google Maps. © 2011 Google - Map data © 2011 Tele Atlas

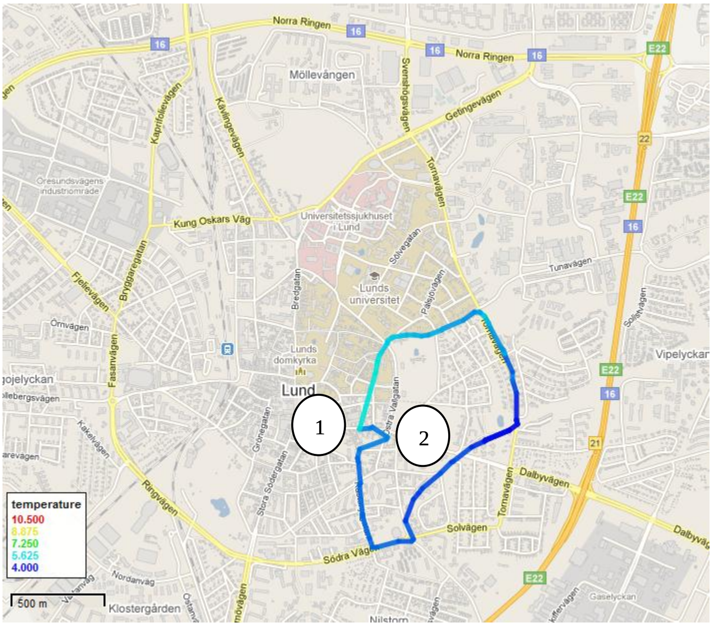

Figure 4.8: Sensor 2 road in experiment 2. Data overlayed on Google Maps. © 2011 Google - Map data © 2011 Tele Atlas

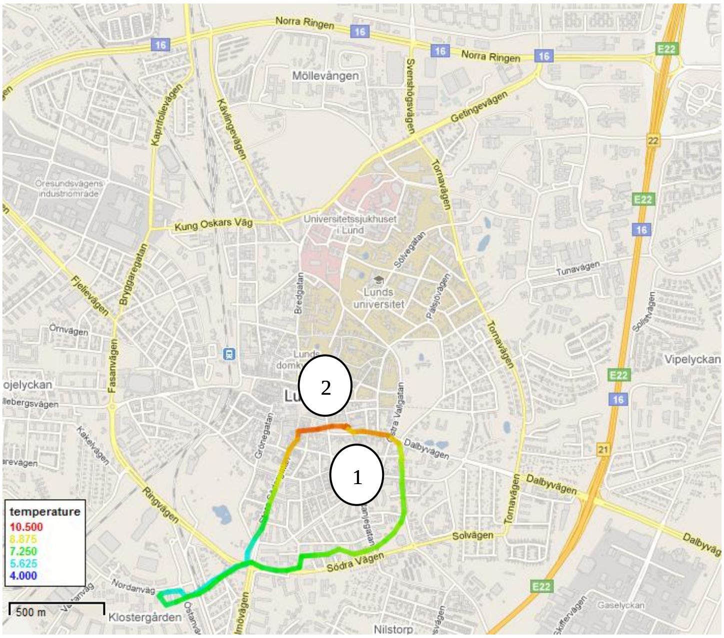

Figure 4.9: Sensor 3 road in experiment 2. Data overlayed on Google Maps. © 2011 Google - Map data © 2011 Tele Atlas

As the maps show, sensors 1 and 2 started at about 5.6 ºC. A while later, their temperatures were decreasing to 4 ºC.

The sensor 3, which was partially covered with the hand, showed a higher temperature than sensor 1 and 2. On the closest zone from Mårtenstorget, it registered 9.5 ºC whereas on the south part of Lund it registered about 7 ºC.

It is worth mentioning that the weather station on E-Huset recorded 3.8 ºC of temperature during the experiment. Thus, all the values from collected samples were above those of the station.

4.2 Comparison with a weather station

In this section the difference in temperature between a weather station and the data gathered by a sensor was studied.

The geographic area where the information was compared was a circle with centre on the weather station in E-Huset (John Ericssons Väg) with a radius of about 200 m.

Now the resultant map is shown:

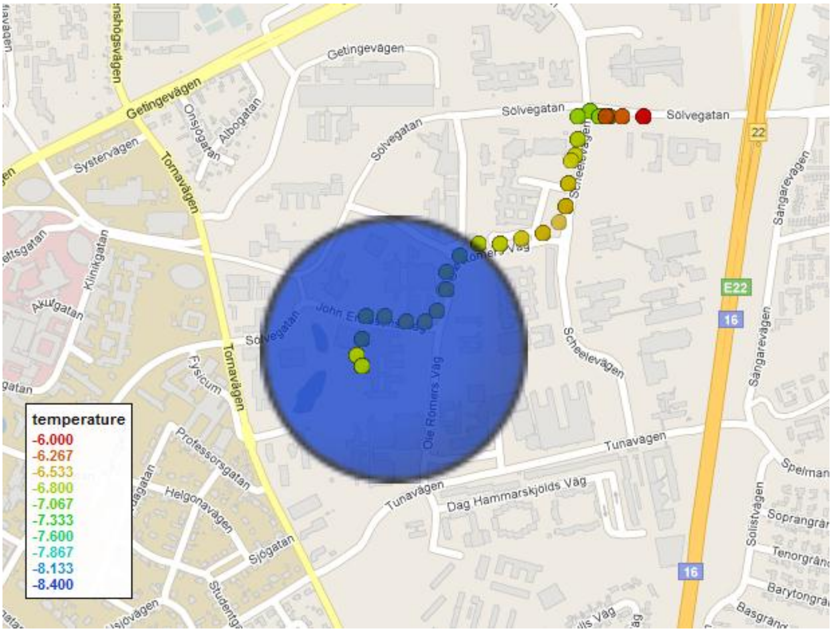

Figure 4.10: Comparison map. Data overlayed on Google Maps. © 2011 Google - Map data © 2011 Tele Atlas

The station registered -8.4 ºC and as it can be seen the difference in temperatures between the sensor and the station inside the circle is about 2 ºC. Owing to the fact that outside the circle the difference was 2.4 ºC, the sensor waste heat did not allow it to stabilize the measurement, since every single temperature sample has a temporal variability (they were taken in different moments). There is also a spatial variability that especially affects in urban areas.

A part from that, there are two samples with a more intense colour inside the circle. This is because the sensors were still, took more than one measurement on the same geographic location and therefore, its dots in the map were painted with more intensity.

4.3 Interpolated map

In this experiment the static performance of the sensors separated about 40 m was studied. The sensors were placed on the central area in Parentesen (Lund). Previously, the sensors were checked to be stabilized at room temperature.

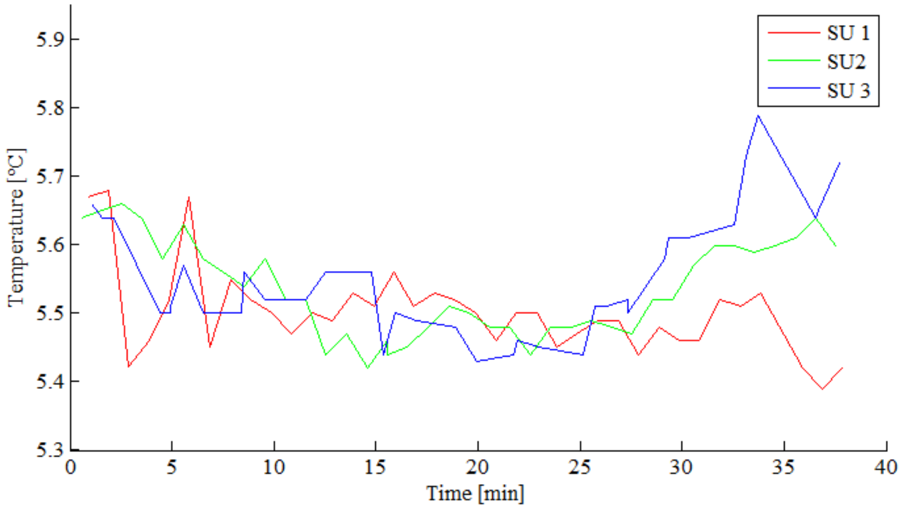

Figure 4.11: Temperature measurement with three sensors fixed

Although during most of the time the three sensors registered temperatures ranging from 5.4 ºC and 5.6 ºC, the maximum variation was 0.4 ºC after 40 min.

To enhance spatial resolution of collected data, a temperature surface has been created using an interpolation process. At one particular moment, with three values of temperature (one for each sensor) it is made a plane equation. Each point on this plane is an estimate of the temperature on a geographic point where no measure was taken.

In the next picture it is shown a surface created at a specific time:

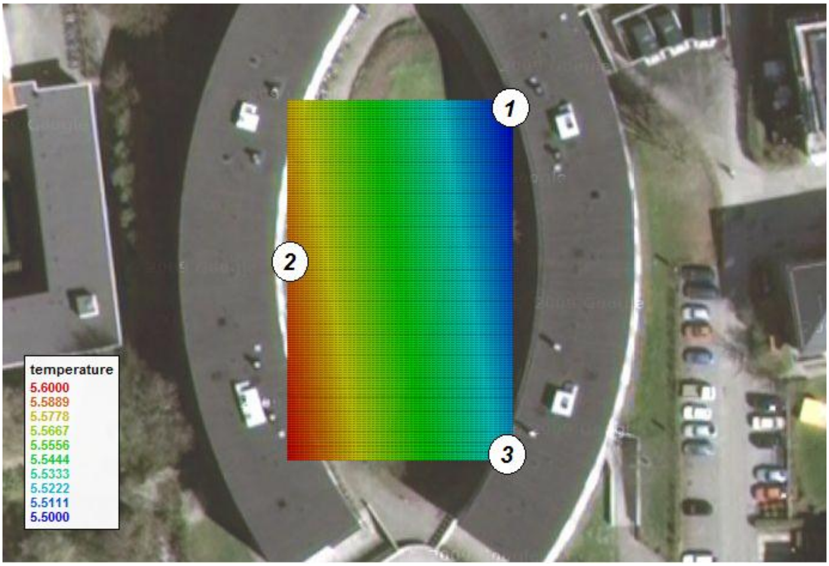

Figure 4.12: Temperature interpolated surface. The numbers indicates the position of the sensors. Data overlayed on Google Maps. © 2011 Google - Map data © 2011 Tele Atlas

Obviously, using just three sensors, the reliability of the interpolated data is low, but in potential future situations where thousands or tens of thousands users utilize the SU this surface could be much more precise.

4.4 *Inside and outside*

This experiment involved going in and going out of a building with the SU and a mobile phone. The goal is to see how long the sensor takes to stabilize the temperature measurement and to test the building detector.

4.4.1. Static

Before beginning this experiment the sensor was placed on a room at 22 °C for 24 hours. Then the sensor was carried to a balcony in the open air, where it was known that the temperature was about 0 °C. Four hours and 20 minutes later it was brought inside the building again and after three hours and 10 minutes the experiment was stopped.

The resulting graph of the experiment is:

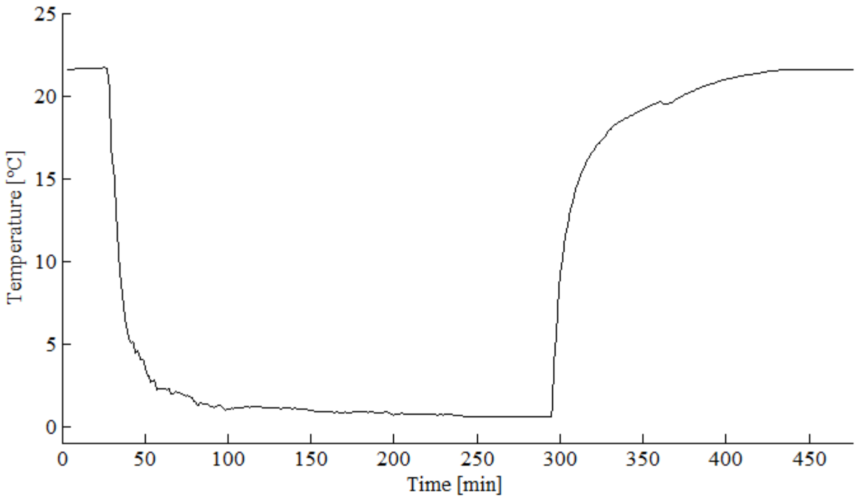

Figure 4.13: Temperature measurement with inside outside static

According to the temperature in Lund [19] it ranged between -0.2 and 0.1 °C when the SU was placed outside. As it can be seen on the graphic, from the minute 100 the measurement was approximately stabilized and close to the real temperature. From that minute to the minute 240 the measurement keeps falling, but only 0.5 °C of difference. Thus, the sensor lasted 1 hour and 10 minutes to reach a difference of 22 °C (from minute 30 to minute 100).

Once the outside temperature was stabilized, the sensor was re-entered to the room with the subsequent rise in temperature. In this case, the temperature change was slower and it took 2 hours and 20 minutes.

It is also worth noting that the opening of the room door caused the small drop in temperature observed at minute 360 where the sensor was placed. The temperature of the room fell slightly but it was not a drastic change in the general shape of the graphic.

Lastly, the building detector was put into practice. The result showed that only the data from the minute 95 to minute 290 was considered as real data.

4.4.2. Dynamic

In this case, the goal is to compare the influence of the person carrying the sensor at a certain speed to the speed of the sensor to stabilize the measurement. To do this, two experiments were only focused to the outdoors behaviour, since inside the buildings the data would be discarded anyway.

4.2.2.1. Walking

The sensor was placed on a room and then it was carried hung on the hand (to interfere in the measures as little as possible) walking for 40 minutes.

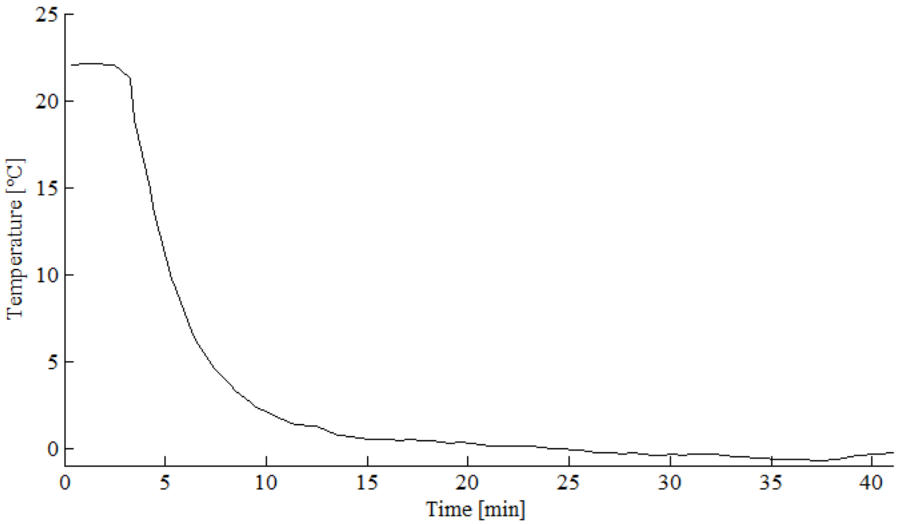

Figure 4.14: Temperature measurement inside outside walking

Comparing this graphic to the previous one it can be seen that the sensor stabilizes faster the measurement if it is in movement. In this case, the collected temperature stabilizes in about 40 min, a 43% faster than the previous.

The building detector decided that it could be considered as real data those between the minutes 29 and 32.

4.2.2.2. Riding a bicycle

In this section the speed of stabilization in the measure of the sensor is analysed when it is placed on a bicycle basket riding for 15 minutes. Firstly, the sensor registered 20 ºC inside a building and then was carried on the bicycle. This is the graphic of the temperature:

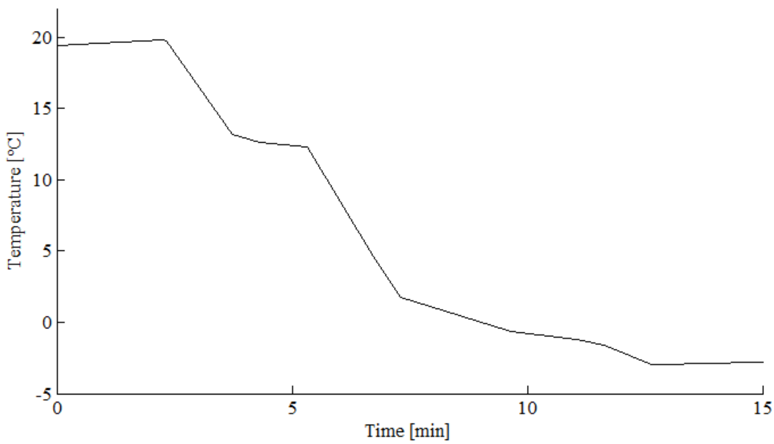

Figure 4.15: Temperature measurement inside outside bike

As the graphic shows, the temperature registered started to decrease at the moment of going out into the street (minute 3). The values kept falling until the minute 13 when the measure was stabilized. Therefore, the sensor lasted 10 min to stabilize the temperature, going from 20 ºC inside a building to -3 ºC outdoors.

According to this test, the measurement reaches its final value faster than walking or being still, as it was expected.

The detector building considered as real data those values collected from the minute 13 onwards.

4.5 Speed test

In this section, the operation of the speed detector was analysed. To do that, three walks have been performed at several speeds:

- 35 min walking
- 25 min riding a bicycle
- 40 min on a bus.

The experiments have the following results:

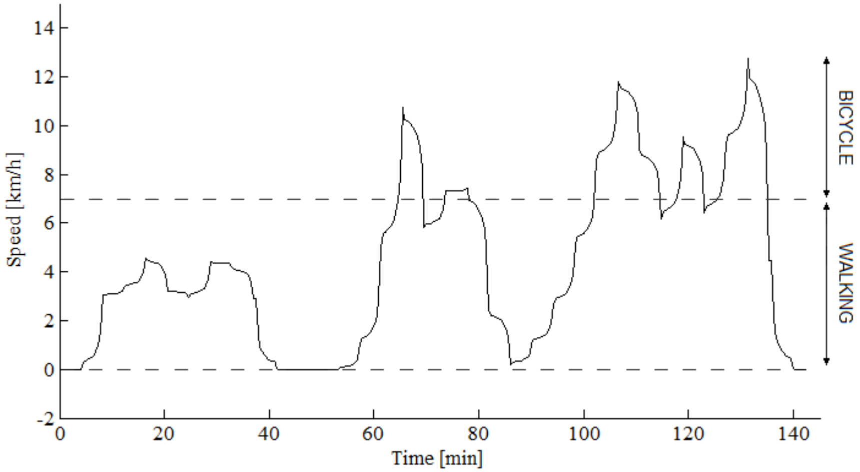

Figure 4.16: Speed testing measurement

The first part of the graphic corresponds with the speed at a walking pace. The values oscillated around 4 km/h, which seems coherent since a usual speed for a human is about 5 km/h [20].

In the middle of the graphic it is shown the speed riding a bicycle, which ranged from 6 to 10 km/h. It is not a high speed as the experiment was made on a day with snowy paths, which made riding a bicycle difficult.

Finally, in the last part of the graphic the bus speed is represented. This speed is between 8 and 12 km/h. This speed is below the real one because the bus stopped frequently due to bus stops and traffic lights. Obviously, although the bus stopped, the sensor kept recording measures and as the speed is calculated with temporal averages. In this last case it was detected a bicycle instead of a bus as the means of transport.

Chapter 5

Conclusions and future work

Once all the experiments are done, a result study to try to conclude qualitatively whether they have been consistent or not is done. For future improved performance, strengths and weaknesses of the system must be identified and quantified. This section will try to answer the following questions:

- Can this type of sensor help to improve the collection of data to be used in better understanding and description of climate change?
- What changes are needed to adapt the system to different applications?
- What scalability complications can occur in the future if there are a large number of sensor units?
- Do the algorithms always work? In all conditions?
- What are the strengths and weaknesses of the sensor unit?
- Will people take part in measuring? Will they do it in optimal conditions?
- Can the existing application be adapted to other mobile phone operating systems?

5.1 The sensor

The main sensor unit handicap is the inertia. In cases where the temperature change is abrupt, the SU cannot detect the new values instantly. At first, the design of the box was completely closed. It was improved by adding holes that increased ventilation, but as the experiments showed it is not enough. To improve this aspect, new prototypes to allow better ventilation for all the integrated sensors should be made and test them properly, like the new suggested prototype in the MMM thesis [2]. It is maybe the most important improvement to be done in the future.

On the other hand, once the taken values are stabilized, they look like the real ones. The only factors to bear in mind are the several offsets found (see section 4.1) due to a calibration error. In the future, a more efficient way of calibration via both software and hardware should be developed.

Finally, since the experiments should be done with a SU and a mobile phone, the main challenge is to implement a built-in SU in the mobile device, which would make it easier to take measurements. Due to the sensor size, one problem would be to create a mobile phone with the usual dimensions but with all the sensors incorporated. Then, studies will be required regarding the heat inertia (because sensors would not probably be properly ventilated) to minimize the heat from the rest of mobile phone electronic components.

5.2 Android application

In general, the application performance is good. All the menus work correctly and the possibility to easily change the sampling time offers great flexibility to the user when taking measurements. Moreover, the gathered information can be stored in the mobile phone memory if there is no Internet access, which makes the execution of the tests possible anywhere in the world.

A part from that, a great improvement to the application would be to automatically reconnect the sensor and the mobile phone in case of Bluetooth connection loss. Occasionally, during some experiments there was a Bluetooth failure without any notification to the user (if the screen was off). Some data gaps appeared consequently, until the user reconnected the system manually.

Lastly, although Android is one of the most used operating system for mobile phones, it is possible to develop it for many others operating systems such as iOS, Windows Phone, Symbian or Maemo. This would reach a greater number of potential users worldwide.

5.3 Server and database

Three sensors have been used to perform the tests. Both server and database operation are good. There is no maximum transmission file size and besides, TCP protocol is responsible for ensuring that the information reaches its destination. However, in a future where there may be thousands of users on the system it may be needed a more powerful server to attend all connection requests.

For that reason, the amount of traffic that would support the system for the following conditions has been calculated:

- Capacity: 10.000 users
- Experiments length: 30 minutes per day
- Interval: 1 measurement per minute

The traffic that should support the server would be 83.7 GB per month and the database capacity would be 1.39 GB per month, as show the formulas in appendix C.2. To reduce this value, in the future some ways to decrease the size of information in each measurement would be:

- For transfers in which some fields are constant in all measures (as mobile phone or sensor name), an improvement would be not to send the redundant information constantly.
- Furthermore, the name of XML tags could be reduced on the template to decrease the file size.
- Finally, the file could be compressed before the sending (in a .zip file for instance). Then, the server would be responsible for unzipping it to have the original XML file.

5.4 Quality assessment

The gathered information quality depends to a large extent on the environment to be analysed. In some cases, the influence of people taking measures and the particular situations in which they are (being inside or outside a building or having the sensor stabilized) affect the result of the measures significantly.

The performance of the detectors is now analysed:

- The calculations of the speed detector are not 100% reliable due to:

  - ✓ There are some situations where it is hard to decide which vehicle has been used. These situations can be running quickly vs. riding a bicycle slowly or riding a bicycle quickly vs. being on an urban bus for example.

  - ✓ The second factor resides in the error in calculating the covered road between the instants when two consecutive measures are taken. If this path is not a straight line, there will be an error associated to the speed, since it is calculated by subtracting two GPS positions. For instance, if a bus turns a corner, it is following a longer road than the road which can be detected by the mobile. The increase of the measurement interval affects to this error. The longer the interval is, the bigger error will be introduced.

- Regarding the building detector, the influence of the mobile phone GPS chipset is important when deciding whether a user is inside a building or not. Therefore, the decision threshold of the detector may vary in different types of mobile phones. In addition, the stabilization threshold of the sensor could be modified if more reliable data is required. One improvement that would help this detector is to use a spatial database with building information.

- Finally, comparisons between stations and nearby sensors can vary depending on the chosen radius. Depending on the accuracy needs, these variations may be higher or lower. An improvement in this aspect, which should be analysed in the future, is the possible comparison of meteorological variables between mobile phones directly.

5.5 Experiments

The experiments were aimed to empirically test the system performance in real situations. They all have been done in the context of temperature differences indoors and outdoors, since the thesis was developed in Lund in wintertime.

It was verified (see section 4.1.1) that when the conditions under which measurements are taken are optimal, the difference between the values collected by the sensors and the data from weather stations or Internet is very low. It is basically an offset in both temperature (figure 4.1) and atmospheric pressure (figure 4.3). Humidity (figure 4.2) is the one that suffers most noticeable variation. To reduce this error in the measurements, a better calibration of the humidity sensor is needed. On the other hand, situation where many sensors register the same data, which are different from a weather station, may indicate that this station needs to be calibrated instead of the sensors.

On the other hand, in the experiment in section 4.1.2, the influence of the person carrying the sensor was evident. If the sensor holes are partially covered, the ventilation is lower and the measures are modified with this kind of design.

Finally, the realization of surfaces with atmospheric values opens up a wide range of possibilities in different applications. The more users the system has, the bigger efficiency the interpolation spatial resolution will have. An interpolation in time was not considered due to the fact that the used frequency in most experiments was one sample per minute.

As the experiments show, high spatial resolution provided by sensors like these could be interest in areas with a lot of people. It may provide local/regional information for various purposes such as microclimate studies on urban environment or agriculture monitoring. Through a future development of both SU and system expounded, the poor resolution of meteorological data in some parts of the world would be improved and thus.

To sum up:

1. The design of the server was successfully made without problems. There is no size limitation for file transfers.

2. The database works quickly with three sensors as well. Maybe in a future with a high number of sensors the query speed will decrease.

3. As the system was designed considering the portable sensor and the mobile phone as the main source of information, the thesis have not focused on collecting data from meteorological websites and weather stations.

4. The last step was the verification of the collected data series with the presented algorithms.
   - Both length and gap detectors works properly in all situations.
   - The speed detector works properly but has some limitations on the speed calculations due to a road error.
   - The building detector has a high dependence on the GPS chipset. In our tests, the HTC Desire had a better performance processing the GPS signals than the HTC Legend.
   - The last algorithm is the one that compares the data with near weather stations. This algorithm depends on the chosen radius and the difference in the altitude with the sensors.

References

[1] Peter A. Stott and Peter W. Thorne, *How best to log local temperatures?*, Nature, Vol 465, 13 May 2010

[2] Fredrik Persson and Tobias Svahn, *Mobile Meteorological Measurements*, Lund University, 6 Dec, 2010

[3] U.S. Enviromental Protection Agency, Greenhouse Gas Emissions, http://www.epa.gov/climatechange/emissions/index.html

[4] OMM, Cambio Climático 2007, http://www.ipcc.ch/pdf/assessment-report/ar4/syr/ar4_syr_sp.pdf

[5] World Meteorological Organization, Information for climate action, http://www.un.org/climatechange/pdfs/bali/wmo-bali07-5.pdf

[6] World Meteorological Organization, *Second Experts' Symposium on Multi-Hazard Early Warning Systems (MHEWS – II)*, http://www.wmo.int/pages/prog/drr/events/MHEWS-II/Documents/Doc-3-MHEWS-II%20Information%20Note.pdf

[7] Reto Knutti and GabRiele C. HeGeR, *The equilibrium sensitivity of the Earth's temperature to radiation change*, http://www.iac.ethz.ch/people/knuttir/papers/knutti08natgeo.pdf

[8] WiseGEEK, What Is a Web Crawler?, http://www.wisegeek.com/what-is-a-web-crawler.htm

[9] World Meteorological Organization, Official Observations. Official Forecasts, http://worldweather.wmo.int/

[10] NOAA, Introduction, http://www.ncdc.noaa.gov/oa/wdc/index.php

[11] HTC, HTC Desire, http://www.htc.com/es/product/desire/specification.html

[12] HTC, HTC Legend, http://www.htc.com/es/specification.aspx?p_id=313

[13] Bluetooth, Basics, http://www.bluetooth.com/English/Technology/Pages/Basics.aspx

[14] Maps gps info, GPS Accuracy - How Accurate is it?, http://www.maps-gps-info.com/gps-accuracy.html

[15] Qualcomm, Snapdragon Chipsets, http://www.qualcomm.com/products_services/chipsets/snapdragon.html

[16] Qualcomm, MSM Chipsets, http://www.qualcomm.com/products_services/chipsets/mobile_processors.html

[17] GPS review, Assisted GPS, http://www.gpsreview.net/a-gps/

[18] GPS Visualizer, http://www.gpsvisualizer.com/

[19] Vackertväder.se, http://www.vackertvader.se/väderstation/lund-n/jämför

[20] Wikipedia, Kilómetro por hora, http://es.wikipedia.org/wiki/Kilómetro_por_hora

Appendix A

Example of an XML code

This is an example of a transmitted file composed by three XML codes:


```
<data>
<source>3</source>
<phoneModel>HTC Legend</phoneModel>
<sensor_mac>00:06:66:05:04:46</sensor_mac>
<day>2011-1-26</day>
<time>09:38:19</time>
<su_version>69</su_version>
<sht_model>11</sht_model>
<sht_version>4</sht_version>
<position_latitude>55.69898843765259</posistion_latitude>
<position_longitude>13.199751377105713</posistion_longitude>
<position_altitude>98.0</position_altitude
<position_accuracy>12.0</position_accuracy>
<temperature>19.239999999999995</temperature>
<humidity>32.20865845000001</humidity>
<bmptemp>19.1</bmptemp>
<pressure>100508</pressure>
<lightSensor>The Android Open Source Project CM3602 Light sensor
1</lightSensor>
<light>0.0</light>
<accelSensor>The Android Open Source Project BMA150 3-axis Accelerometer
1</accelSensor>
<accelerationx>0.10896278</accelerationx>
<accelerationy>0.23154591</accelerationy>
<accelerationz>9.765789</accelerationz>
<proximitySensor>The Android Open Source Project CM3602 Proximity sensor
1</proximitySensor>
<proximity>1.0</proximity>
</data>


<data>
<source>3</source>
```

<phoneModel>HTC Legend</phoneModel>
<sensor_mac>00:06:66:05:04:46</sensor_mac>
<day>2011-1-26</day>
<time>09:39:14</time>
<su_version>69</su_version>
<sht_model>11</sht_model>
<sht_version>4</sht_version>
<position_latitude>55.69889724254608</posistion_latitude>
<position_longitude>13.199772834777832</posistion_longitude>
<position_altitude>85.0</position_altitude
<position_accuracy>4.0</position_accuracy>
<temperature>19.299999999999997</temperature>
<humidity>32.08164528200001</humidity>
<bmptemp>19.2</bmptemp>
<pressure>100518</pressure>
<lightSensor>The Android Open Source Project CM3602 Light sensor
1</lightSensor>
<light>0.0</light>
<accelSensor>The Android Open Source Project BMA150 3-axis Accelerometer
1</accelSensor>
<accelerationx>-0.0</accelerationx>
<accelerationy>0.27240697</accelerationy>
<accelerationz>9.765789</accelerationz>
<proximitySensor>The Android Open Source Project CM3602 Proximity sensor
1</proximitySensor>
<proximity>1.0</proximity>
</data>

<data>
<source>3</source>
<phoneModel>HTC Legend</phoneModel>
<sensor_mac>00:06:66:05:04:46</sensor_mac>
<day>2011-1-26</day>
<time>09:40:22</time>
<su_version>69</su_version>
<sht_model>11</sht_model>
<sht_version>4</sht_version>
<position_latitude>55.69892406463623</posistion_latitude>
<position_longitude>13.199762105941772</posistion_longitude>
<position_altitude>88.0</position_altitude
<position_accuracy>3.0</position_accuracy>
<temperature>19.259999999999998</temperature>
<humidity>32.1773591445</humidity>
<bmptemp>19.1</bmptemp>
<pressure>100512</pressure>
<lightSensor>The Android Open Source Project CM3602 Light sensor
1</lightSensor>

<light>0.0</light>
<accelSensor>The Android Open Source Project BMA150 3-axis Accelerometer 1</accelSensor>
<accelerationx>0.040861044</accelerationx>
<accelerationy>0.23154591</accelerationy>
<accelerationz>9.80665</accelerationz>
<proximitySensor>The Android Open Source Project CM3602 Proximity sensor 1</proximitySensor>
<proximity>1.0</proximity>
</data>

Appendix B

Database table specification

These are the fields that compose the table that stores all the information sent by all the sensor units. The name of this table is *Incoming_data*.

Field	Type	Null	Key	Default	Extra
Source	int(11)	YES		NULL	
IMEI	bigint(20)	YES		NULL	
Day	date	YES		NULL	
Time	time	YES		NULL	
Position_latitude	decimal(20,10)	YES		NULL	
Position_longitude	decimal(20,10)	YES		NULL	
Position_accuracy	float	YES		NULL	
Position_altitude	decimal(20,10)	YES		NULL	
Bmptemp	float	YES		NULL	
Temperature	float	YES		NULL	
Et	float	YES		NULL	
Humidity	float	YES		NULL	
Eh	float	YES		NULL	
Pressure	float	YES		NULL	
Ep	float	YES		NULL	
Light	float	YES		NULL	
El	float	YES		NULL	
Lightsensor	varchar(100)	YES		NULL	
Accelsensor	varchar(100)	YES		NULL	
Accelerationx	float	YES		NULL	
Accelerationy	float	YES		NULL	
Accelerationz	float	YES		NULL	
Proximitysensor	varchar(100)	YES		NULL	
Proximity	float	YES		NULL	
Comment	varchar(100)	YES		NULL	
Radiation	float	YES		NULL	
Wind	float	YES		NULL	
Direction	float	YES		NULL	
Sensor_mac	varchar(20)	YES		NULL	
Phonemodel	varchar(20)	YES		NULL	

This is the visual appearance of the table storing one XML code:

```
-------------------+-------------------+---------+-------------+------+----------+
------+----------+------+-------+------+------------------------------------------
-------------+---------------------------------------------------------------+----
-----------+---------------+---------------+--------------------------------------
---------------------+-----------+---------+-----------+------+-----------+-------
------------+------------+
| Source | IMEI | Day        | Time     | Position_latitude | Position_longitude |
 Position_accuracy | Position_altitude | Bmptemp | Temperature | Et   | Humidity |
 Eh   | Pressure | Ep   | Light | El   | Lightsensor                              
             | Accelsensor                                                   | Acc
elerationx | Accelerationy | Accelerationz | Proximitysensor                      
                     | Proximity | Comment | Radiation | Wind | Direction | Sensor
_mac        | Phonemodel |
+--------+------+------------+----------+-------------------+--------------------+
-------------------+-------------------+---------+-------------+------+----------+
------+----------+------+-------+------+------------------------------------------
-------------+---------------------------------------------------------------+----
-----------+---------------+---------------+--------------------------------------
---------------------+-----------+---------+-----------+------+-----------+-------
------------+------------+
|      3 | NULL | 2011-01-26 | 10:00:35 |     55.7110574000 |      13.2098227300 |
                60 |              NULL |     3.5 |        7.38 | NULL |  51.4822 |
 NULL |   100158 | NULL |     40 | NULL | The Android Open Source Project CM3602 Li
ght sensor 1 | The Android Open Source Project BMA150 3-axis Accelerometer 1 |    
 -0.953424 |       3.75922 |       9.61597 | The Android Open Source Project CM360
2 Proximity sensor 1 |         1 | NULL    |      NULL | NULL |      NULL | 00:06:
66:05:04:46 | NULL       |
+--------+------+------------+----------+-------------------+--------------------+
-------------------+-------------------+---------+-------------+------+----------+
------+----------+------+-------+------+------------------------------------------
-------------+---------------------------------------------------------------+----
-----------+---------------+---------------+--------------------------------------
---------------------+-----------+---------+-----------+------+-----------+-------
------------+------------+
```

Appendix C

C. Formulae

C.1. Speed formulae

From GPS positioning variations, which are ΔLat (Latitude) [°], ΔLong (Longitude) [°] and ΔAlt (Altitude) [m], distance is calculated as:

$$\mathbf{lat\_m = \Delta lat \cdot 60 \cdot 1852\ [m]} \tag{C.1. 1}$$

$$\mathbf{long\_m = \Delta long \cdot cos(\frac{lat1 \cdot \pi}{180}) \cdot 60 \cdot 1852\ [m]} \tag{C.1. 2}$$

$$\mathbf{alt\_m = \Delta alt \cdot 0.3048\ [m]} \tag{C.1. 3}$$

$$\mathbf{distance = \sqrt{lat\_m^2 + long\_m^2 + alt\_m^2}\ [m]} \tag{C.1. 4}$$

where lat_m is Δlat expressed in meters and Δlat=lat2-lat1
long_m is Δlong expressed in meters
alt_m is Δalt expressed in meters

Finally, the speed is calculated in kilometres per hour using a conversion factor:

$$\mathbf{s = \frac{\Delta distance(m)}{\Delta time(s)} \cdot \frac{1\ km}{1000\ m} \cdot \frac{3600\ s}{1\ h} \left[\frac{km}{h}\right]} \tag{C.1. 5}$$

C.2. Traffic formula

Traffic per month of the system:

$$\boldsymbol{traffic = 10.000\ users \cdot \frac{30\ days}{1\ month} \cdot \frac{30\ min\ measuring}{1\ day} \cdot \frac{1\ measurement}{1\ min} \cdot \frac{9.3\ kB}{1\ measurement} = 83.7\ GB/month} \tag{C.2. 1}$$

Database capacity:

$$\boldsymbol{capacity_{db} = 10.000\ users \cdot \frac{30\ days}{1\ month} \cdot \frac{30\ min\ measuring}{1\ day} \cdot \frac{1\ measurement}{1\ min} \cdot \frac{166\ bytes}{1\ measurement} = 1.39\ GB/month}$$