Educational Microclimate Mesonet System Design

James Robertson[1], Jianhua Yang[1(✉)], W. Scott Gunter[2], Peter Keres[1], Thomas Wingate[1], and Anthony Obando[1]

[1] TSYS School of Computer Science, Columbus State University, 4225 University Ave., Columbus, GA 31907, USA
{robertson_james5, yang_jianhua}@ColumbusState.edu
[2] Department of Earth and Space Science, Columbus State University, 4225 University Ave., Columbus, GA 31907, USA
gunter_william@ColumbusState.edu

Abstract. Through the collaborative efforts of Columbus State University (CSU), Oxbow Meadows Environmental Learning Center, and Columbus Water Works, the CSU Weather Station was constructed in Fall 2018. This paper presents an overview of the construction efforts, instruments, real-time network, and database used on the tower as well as the collaborations that allowed for the development of the website. A statistical summary of the validation efforts and highlights of interesting weather events are also presented. The CSU Mesonet includes Tower station, Data server, Database, and Web server. In this paper, we focus on not only how the data is collected, delivered from the tower to the data server, as well as from data server to the database server, but also how the weather data is browsed via the web server and database server.

1 Introduction

1.1 What Is Mesonet?

Mesonet is part of a more sophisticated "mesonetwork", which collects standard atmospheric variables of temperature, relative humidity, air pressure, wind speed, wind direction, rainfall, and solar radiation. The collection rate is adjustable such that the data can be used for either research projects or the development of educational materials. These real-time collection rates will be especially critical for the development of value-added educational tools, networks, system designs, and training. These education and training tools will be used in order to better serve the needs of the local, national, and international communities.

1.2 The Significance of Mesonet

Mesonet plays a significant role in the decision-making process of Emergency Management which makes life and death decision during severe weather. Mesonet allows pilots to have better weather information during landing and taking off, the most critical stage of flight, as well as drone delivery and air taxis. It is also useful for farmers

L. Barolli et al. (Eds.): WAINA 2020, AISC 1150, pp. 24–33, 2020.
https://doi.org/10.1007/978-3-030-44038-1_3

choosing when to plant and harvest their crops. These are just some of the applications that the Mesonet plays upon our lives and the role that it plays in our society. By understanding better network design will allow us to determine the best way to create the CSU Mesonet Network that will be used as a research tool, education tool, training tool, and decision-making tool.

"Mesoscale surface observations are vital data sources for applications in many different meteorological subfields, including operational forecasting, wind power management, transportation safety, wildfire management, dispersion modeling, and defense applications. Two recent reports (National Academy of Sciences 2009, 2010) recommend that existing and future mesoscale observations be integrated into a network of networks. The heterogeneous nature of the available mesoscale surface observing networks within the United States (e.g., varying sensor quality, maintenance and reporting practices, and siting) can limit their potential benefits" [1].

A critical recommendation in both reports is to improve the metadata that define the sensor and station characteristics within the aggregated networks. Users of the national network would then be able to select the types of stations that meet their specific needs [1].

1.3 Mesonet in USA, and Its Roles

The Mesonet was created out of an event that happened in 1982. The Mesonet came out of the discovery that there needed to be better tools for scientist to expand upon weather data for a flood-warning system and farmers as a whole. When Oklahoma State University (OSU) and Oklahoma University (OU) joined forces, they came to the realization that one system would fill both universities' needs. In 1990, the project was funded with two million dollars from oil funds [2]. Once the system was fully funded, it expanded to 120 automated Mesonets throughout Oklahoma [3].

We are designing the Mesonet in a way to relay data in real-time and send the data in the most efficient way to the web server, subsequently being displayed on a browser site. Although there are currently 1,698 Mesonets in service, we still need to develop better, more time and energy efficient systems. It is widely accepted that 'as close to real time' is acceptable within a system design. The design that we are proposing is to make it real-time. Designing the network of networks and creating a real-time Mesonet will allow for better and faster decision making to take place.

1.4 Introduction to CSU Mesonet Project

The CSU Mesonet consists the following elements: air temperature sensor, relative humidity sensor, air temperature sensor, wind speed and direction sensor, barometric pressure sensor, rainfall measurement sensor, solar radiation sensor, soil temperature sensor, 30-foot tower, rf451-1 Watt radio, and a CR3000 measurement and control data logger.

CSU Mesonet is solar power-driven with a 55-watt solar panel and a battery that stores enough power for ten days of backup during inclement weather. Once data is received from each sensor, the tower sends a signal with the data to the CR3000 data logger. The data logger then stores and fetches the data. The data will be transmitted via narrow-band UHF/VHF to the receiver antenna attached to the top of the data receiver building. This data is then relayed to Windows Server, also called data server. Once the data is stored onto the server, it is transferred to an SQL database using customized syntax in order to automate the data transfer. Once data is transferred to the database, the information is transmitted to CSU's web server. The web server is designed to be secure and is managed by Columbus State University. The website and data storage will be located on two separate parts of the server, which will allow us to access the HTTP site. CSU Weather Station Network is configured so that once the wind sensor recognizes wind speed and wind direction, it will send the data to the data logger. The data logger uses radio signal to transfer the data to the data server. CSU Mesonet transfers weather data from the tower station to the web server using Transmission Control Protocol / Internet Protocol (TCP/IP) [4]. In the following section, we will discuss in more details how the weather data is transferred.

2 The Architecture of CSU Mesonet

CSU Mesonet contains three primary components: data collection, data storage, and data browsing. Data collection includes the tower station, and the data server. The tower station has twelve different types of sensors assembled to collect wind speed, air temperature, air humidity, wind direction, etc., the weather information. The weather data collected is stored in the memory card of the data logger that is attached to the tower. The data stored in the data logger can be transferred to the data server, which sits 300 feet away from the tower station, using radio signal via RS-232 port. The data server connects to the Internet. A TCP/UDP socket is used to transfer the data from the data server to the database server, which is located at CSU's main campus. The weather data can be accessed by any type of browser through the web server.

2.1 Data Collection Tower Station

CSU Mesonet tower station is located in Oxbow Meadows, Fort Benning, Georgia. Oxbow Meadows is a center opened in 1995 as a collaboration among CSU, Columbus Water Works, and the City of Columbus. Its Environmental Learning Center can provide CSU students with exhibits, natural history displays, nature trails, and can offer formal and informal educational programs about the ecology and natural history of the region. Oxbow Meadows has an open, champaign country which is suitable to monitor the information about weather, such as wind speed at 2-m, and 10-m, wind direction, air temperature at 1.5-m, 2-m and 10-m, solar radiation, 1.5-m relative humidity, barometric pressure, rain gauge, leaf wetness, soil moisture, and soil temperature.

Currently the tower station has six sensors installed, which detect temperature, humidity, wind speed, wind direction, barometric pressure and rainfall, respectively. Based on the collected weather information, scalar mean wind speed, resultant mean wind speed, resultant mean wind direction, standard deviation of resultant wind direction, maximum one second wind speed, and the time of maximum gust can all be easily obtained. All the above information is stored in the data logger in a text file containing multiple records. After each minute, a new record is generated and written to the data file.

Each record in the data file contains thirteen fields/columns. Column 1 represents the time of the record in the format of YYYY-MM-DD HH:MM:SS with time in Eastern Standard Time (EST). Column 2 is the record number. Column 3 records the minimum battery voltage in Volts. Column 4 stores the temperature degrees Celsius. Column 5 gives the relative humidity percentage around the tower station. Column 6 indicates the scalar mean wind speed as meters per second. Column 7 reflects the resultant mean wind speed as meters per second. Column 8 reveals the resultant mean wind direction in degrees Celsius. Column 9 keeps the standard deviation of resultant wind direction in degrees Celsius. Column 10 stores the maximum one second wind speed as meters per second. Column 11 represents the maximum time with the same format as Column 1. Column 12 records the barometric pressure in hectopascals (hPa). Column 13 stores the rainfall as inches.

CSU, in partnership with Oxbow Meadows, selected a 300 square-foot plot of land designated for the weather station tower. The tower is aluminum and is anchored to a concrete slab. The CSU weather station stands at thirty feet tall, and is held in place by guide wires. Installed on the tower are sensors, a solar panel, CR 3000 data logger, and a battery box with a battery at the base of the tower. The sensors are hard-wired into the CR 3000 data logger. At the base of the tower, located ten feet away is a Campbell Scientific Precipitation Sensor. The Oxbow tower is off-grid and operated by batteries and solar panels.

The six sensors attached to the weather station that collects and feeds data to the data logger are:

1. Temperature – The temperature sensor is mounted twenty feet above the ground.
2. Wind speed – Mounted ten feet above the ground
3. Humidity
4. Wind direction
5. Barometric pressure
6. Precipitation sensor

Each sensor gathers and collects data that is sent to the data logger. This data is transferred via radio signal to the data server, then downloaded into the database server. Once the data is collected into the database, the website will connect to the database and display the collected data in real-time. This allows any individual to see the real-time data and make value added decisions from the data collected. These decisions are used to improve the local and international communities.

2.2 Data Server

The weather data collected at the tower station will be transferred to the data server via radio communication through RS-232 port of the data logger. The data server, located 300 feet away from the tower station, connects to the Internet via AT&T support.

The data server is a server that architecturally resides in between the tower station and the database server at CSU. It receives the weather data information from the tower station via an antenna using radio signals and then forwards the data to the database server using TCP/UDP socket communication. The data server acts as both a receiver and a sender by running two processes concurrently with one receiving data and writing to a temporary file, and another one reading data from the temporary file and sending the data to the database server via the Internet.

2.3 Database Server

The database server receives and stores the data received from the data server in an MS-SQL database. It also provides the data the web server for browsing. The server runs a TCP/UDP socket code to communicate with the data server. It runs Windows 2018 Server Operating System, and MS-SQL Database Management System (MS-SQL DBMS). The communication between the database server and the web server is through the Internet using ASP.NET CGI code.

2.4 Web Server

The web server provides an interface for users to access and browse the weather data that is stored in the database server. The web server can get a request from a browser and forward the request to the database server. The database server searches its database for data that matches the received request and returns the results to the web server.

The web server forwards the results received from the database server to the client-side browser. The browser can interpret the response received from the web server and display it in a readable format.

There are primarily five different types of popular web servers: Apache, IIS, Lighttpd, Sun Java Web Server, and Jagsaw. They are popular not because their stable performance, but also their low/no cost, and are relatively simple to maintain. Some other unpopular, expensive web servers include Netscape's iPlanet, Bea's Web Logic, and IBM's WebSphere.

The Lighttpd is a free, open source web server that is distributed with the FreeBSD operating system. It is secure, fast, and consumes much less CPU power compared to the previous web servers. This type of web server can run on most popular operating systems, such as Windows, Mac OS X, Linux and Solaris Operating Systems. Sun Java System Web Server is designed for medium and large websites and free, but not open source. Jigsaw server was developed by the World Wide Web Consortium. It is open source, free, and can run on various platforms. Apache HTTP server was developed by the Apache Software Foundation. It is open source and can run on almost all the operating systems. It is the most popular web server in the world. About 60% of the web servers in the Internet run the Apache HTTP server.

The Internet Information Server (IIS) was developed by Microsoft. It is not generally free to use, nor is it an open source system; however, it is free for CSU since we are a member of Microsoft College League. It can run on Windows NT and Server system. In our Mesonet, we select to use IIS because it is tightly integrated with Windows OS, high performance, and easy to administer.

3 Collected Data Transfer

The collected weather data at the tower station can be stored temporarily at the data logger, then sent to the data server using RF451 900 MHz 1 W Spread-Spectrum Radio. The data server stores the received data to a text file, and then sends it to the database server located at CSU via TCP/UDP socket communication. The database server writes the received weather data to a database table, and also feeds the web server for client browsing. Due to the Internet availability, except the tower station, all the data server, database server, and the web servers connect to the Internet.

3.1 From Tower Station to Data Server via Radio

For the data delivery from the tower station to the data server, we use an RF451 900 MHz 1 W Spread-Spectrum Radio. The RF451 connects through point-to-multipoint wireless network. It establishes a connection to a PakBus located 800 meters from the Mesotower. "The RF451 is a 902 to 928 MHz frequency–hopping spread–spectrum radio" (https://www.campbellsci.com/rf451). The first radio will be connected to the server and be set as the master radio and the second radio will be established as the data logger. The connection link between the radios can be treated as a multi–drop, high–speed connection [5].

"The RF451 consists of a radio module manufactured by FreeWave Technologies and a Campbell Scientific interface board. It reduces susceptibility to RF interference from other spread–spectrum devices by providing user–selectable frequency hopping patterns. Spread–spectrum radios spread the normally narrowband information signal over a relatively wide band of frequencies. This process allows communications to be more immune to noise and other interference" [5].

Powering the radio: "at least two radios are required to create a link. The radio may be powered through the DC barrel connector or via a CS I/O connection. When AC power is available, the 15966 wall charger is commonly used. At remote sites, the RF451 typically is powered through the CS I/O or the 14291 field cable" [4].

Antennas: "Campbell Scientific offers a variety of antennas for this radio. The 14204 is a 0 dB, 1/2 wave omnidirectional whip antenna that connects directly to the radio (no cable required) and can transmit short distances (up to 1 mile). The 15970 dipole antenna includes adhesive for window or wall mounting and a cable for connecting to the radio" [5].

3.2 From Data Server to Database Server via UDP Socket

There are two programs running at the data server. One is to receive data from the tower station and write the data to an intermediate file "WeatherData.dat", and the other is to read data from the file "WeatherData.dat" and send the data to the database server via the Internet. We name the first program as "ReceiveFromTower", and the second one as "SendToDatabase".

The operation of the file "WeatherData.dat" is a reader/writer problem. The program "SendToDatabase" plays a role of reader, and "ReceiveFromTower" plays a role of writer. The writer can write one weather data record to the file "WeatherData.dat" in every 30 s at most, and the reader can read from the file at a comparable frequency. Every time there is only one record read and sent. The writer needs to lock the file "WeatherData.dat" while writing. The reader can open the file in "Sharing" mode without locking it, but it needs to keep a pointer to indicate the last record sent out. The reader works as a client and uses a TCP/UDP socket to send each record to the database server.

The writer feeds the file "WeatherData.dat" at the end. In order to prevent the file from becoming too large to operate, the reader deletes 120 records from the file as long as it contains more than 120 records. The deleted records are written to a daily based "Date_DataLog" log file.

The database server works as a server to receive weather data from the client using a TCP/UDP socket. As long as a new record is received, it is written into the data table of the database. The table needs to be locked while writing.

3.3 From Database Server to Web Server via ASP.NET

The weather data stored in the database server can be accessed by a browser via the web server. The request to browse the weather data is delivered to the web server first. The web server accepts and forwards the request to the database server. The database server runs an SQL procedure to search the weather database table and returns the results back to the web server. The web server sends the response back to the browser, and the browser can interpret the response from the web server and display the requested weather data.

The communication between the web server and the database server is handled under ADO.NET Entity Framework. We use LINQ (Language-Integrated Query) to access MS-SQL Server weather database. LINQ has been integrated into ASP.NET framework. LINQ can access and query a wide variety of data sources. There are three different types of LINQ including (1) LINQ to XML files, (2) LINQ to Objects, and (3) LINQ to ADO.NET. ADO.NET is part of .NET framework that enables us to access data, data services, and data sources. With LINQ and ADO.NET, we can query database related information sets, such as LINQ to Entities, LINQ to Dataset, and LINQ to SQL Server.

We use LINQ to SQL Server to access the weather data stored in the database server from the web server. Since the database server uses MS SQL database, so it is actually LINQ to MS SQL server. It is first creates a layer between the .NET application and the SQL Server database. Then it can turn the database objects, such as tables, views, into .NET objects. Therefore, it is trivial to map database items to objects in the application by using LINQ to SQL.

4 Database Design

A database provides a good approach to store a large amount of information. It makes easy to access the information using a Database Management System (DBMS). In this project, the records gathered from weather station sensors are stored into a database at a selected time interval which is determined by the system administrator. E.g. Every thirty seconds, readings are sent and stored into the database.

Taking the classic approach of manual record keeping, the records could be stored as text files each day in a folder. While this approach could work, it is not practical when considering the vast amount of data that would need to be culled through when accessing the data for scientific use. The flaws using flat text fields include but are not limited to the following:

Speed. Databases use a technique called 'indexing' to optimize its performance. It is a technique allowing users to quickly locate and access data without having to possibly traverse the entire length of the database. Using flat text files would drastically lower performance in this case.

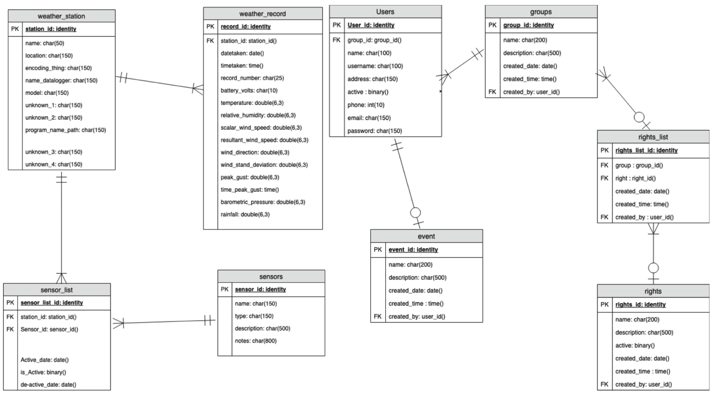
Integrity. Data integrity is the completeness, consistency, and accuracy of data. Databases allow us to enforce rules regarding data storage and formatting. E.g. Weather Station Alpha tracks three sensors: temperature, wind speed, and rainfall. If a fourth item is added, the database will determine that item as an invalid record and will not add it to the database.

Ease of use. Databases allow search and storage of data with a relatively small amount of work from the developer. By using Structured Query Language (SQL), complex searches become very simple. For example, if we needed to access records where rainfall was greater than one inch, the query would be 'SELECT * FROM RECORDS WHERE RAINFALL >1'. To complete the same task using flat text files, we would need to create entirely new methods and functions that interact with those files, which could be time consuming and inefficient.

A database includes two major parts: SQL and DBMS. The DBMS is a lightweight program that interacts with the data in the database. SQL is standardized across multiple DBMS systems, allowing it to be used across multiple different DBMSs in the same manner. The DBMS we will be using in this project is SQLite.

SQLite is a specific type of DBMS called a relational database management system. Relational refers to the ability to construct relations between data sets. It was created in the C programming language and implements most of the standard SQL library. It is also ACID (Atomicity, Consistency, Isolation, and Durability) compliant. We chose SQLite for a few reasons. SQLite is very portable because it is a file-based, serverless system. As a result, time and complexity to configure the database will be minimal. SQLite is a great for any testing or development process because of its ease of use and portable nature.

The whole database designed in this project includes tables of Users, Groups, Rights_list, Event, Rights, Weather_station, Weather_record, Sensors, and Sensor_list. The following model view gives the structure of each table, as well as the relations among the tables. Primary key and foreign keys in each table are also specified.



5 Weather Station Website Architecture

The website is an interface allowing users to access the weather station data. The whole site is designed as a tree structure starting from a root home page. The three-click rule was adopted to guide the website design. From the home page to access any information, it needs no more than three clicks. Dynamic real-time weather data can be browsed directly as long as the home page is loaded up anonymously. Authenticated users can obtain more critical data. Administrator can create a new user. Any anonymous user can access stored historical data by modifying parameters, such as time range, date range, station ID, sensor type, or data type.

The weather data information can not only be displayed in a table format, but also be displayed in a plot, which makes data easily readable and more understandable for end users. The website also contains static web pages to display information about the weather station, sensors, and mesonet project overall and active pages to provide simple weather data analysis. We use C# language to make active pages under the framework of ASP.NET. Active pages allow interaction between the web server and the database server. For example, users can get the highest temperature, fastest wind speed, or a specific humidity from their browsers.

6 Conclusion

This Educational Microclimate Mesonet System has been designed through the collaborative efforts of the Department of Earth and Space Science and the TSYS School of Computer Science of Columbus State University. The Mesonet includes the components of data collection, data transfer, data storage, data processing, and data access and usage. It not only is used for educational purpose for students, but also provides faculty conducting their research on the weather changes in Columbus area. The Mesonet hardware includes tower station sensors, data logger unit, radio transfer unit, data server, database server, and web server. This project can allow the students in the Department of Earth and Space Science to monitor real-time weather data, and study changes in weather. The students in the TSYS School of Computer Science can apply what they learn in classes of computer networking, network socket programming, database design and SQL programming, website design, and web server CGI programming to the Mesonet system design.

References

1. Tyndall, D.P., Horel, J.D.: Impacts of Mesonet observations on meteorological surface analyses, 27 February 2013. https://journals.ametsoc.org/doi/full/10.1175/WAF-D-12-00027.1. Accessed 6 Oct 2019
2. Oklahoma Climatological Survey: What is the Oklahoma Mesonet? (1997) https://www.mesonet.org/files/materials/mesonet.pdf. Accessed 3 Oct 2018
3. Mesonet: About the Mesonet. https://www.mesonet.org/index.php/site/about. Accessed 3 Oct 2019
4. Mombrea, M.: How to fetch a file from FTP and import into SQL server automatically, 18 November 2014. Retrieved from https://www.itworld.com/article/2847292/how-to-fetch-a-file-from-ftp-and-import-into-sql-server-automatically.html
5. Campbell Scientific: RF451 900 MHz 1 W Spread-Spectrum Radio (2019). https://www.campbellsci.com/rf451. Accessed 6 Oct 2018