METEOROLOGY & CLIMATOLOGY GFA OPEN DATA FOR CASTILLA Y LEÓN

Ángel Manuel Guerrero-Higueras[1], Andrés Merino[1], Laura López[1], José Luis Sánchez[1], Vicente Matellán[2], Eduardo García-Ortega[1], José Luis Marcos[1], Lucía Hermida[1]

Universidad de León, Campus de Vegazana S/N, León, España

Abstract

Meteorological data is mainly obtained from forecasting and observation. Forecasting models provide information on the state of the atmosphere in the future, which has to be validated using data from observation systems. These include weather station networks, hail sensors, disdrometers, radiosondes, radar and satellites. The spatial and temporal heterogeneity of the data obtained means that it is very difficult to use it together in research projects.

Currently, there are no data repositories available to the scientific community which attempt to bring all of this information together. Having all of this data in a single, accessible repository would represent a great step forward that would make it easier to develop tools for meteorological risk detection in real time, data assimilation and the real time validation of forecasting models, or simply for providing weather information to the general public.

The Atmospheric Physics Group (GFA) from the University of Léon is gathering meteorological data with the aim of creating a data repository, known as GFA Open Data, which can be used by the scientific community for a variety of purposes. At present, the GFA has established contacts with several risk management bodies in order to include their data in the repository, such as the Duero River Authority (CHD), Ebro River Authority (CHE), Agricultural Technology Institute of Castile-León (ITACyL), the Civil Protection Authority and EUMETSAT. The GFA uses this information in their research projects, such as implementing an algorithm which calculates hail probability using MSG data, as shown in Merino et al. (2013).

The aim of the GFA is to develop a data repository with information from as many sources as possible available to anyone without restriction. At present, there are no repositories like this in Castile-León.

DATA MODEL

Combining data from different sources is a complex task, which requires creating a data model that is capable of storing assorted information. To do so, it is necessary to identify common points in the data, regardless of how they are represented (NetCDF, etc.).

All meteorological data represents the value of a given meteorological variable, such as temperature, relative humidity, etc. at a given time and in a given location. This means we can identify two entities for the data model:

1. *Place*: A set of geographical points characterized by their geographical coordinates, latitude, longitude, and height above sea level.
2. *Variable*: A set of meteorological variables, such as temperature, relative humidity, etc.

The relationship between a certain instance of place and a certain instance for a variable represents a measurement, providing a specific time and a specific value for the meteorological variable. For example, on October 18th 2013 at 12:00 UTC, the temperature in the city of León, Spain was 20ºC.

Each entity or relationship in a data model has a number of attributes that defines each instance. For example, an instance of *place* is defined by its geographical coordinates and its height above sea level. An instance of *variable* is defined by a meteorological variable, such us the temperature, units of measurement (such as ºC), a description, etc. The following sections examine each of the entities in the Open Data model.

Place entity

The first entity required in order to create the data model is *place*. As shown in figure 1, an instance of *place* represents a geographical point characterized by its coordinates, latitude and longitude, and its height above sea level. An instance of *place* may represent a weather station location or a grid point on MSG Data. This is associated with the following entity.

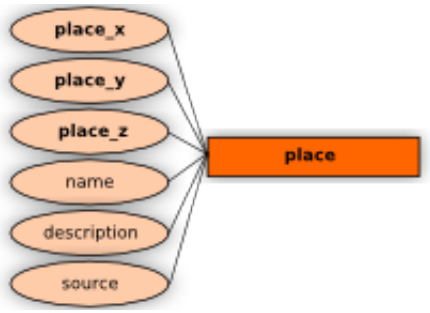

Figure 1: **place entity.**

The attributes that characterize an instance of *place* are the following:

- latitude.
- longitude.
- height above sea level.

There are also other attributes that can complete the information in a given place instance, such as the nearest town, region, province, country, etc.

Source entity

In order to differentiate and group the origin of *place* instances, the entity *source* is used. An instance of *source* represents a group of *place* instances belonging to the same organization or representing the same data type. Currently, these origins can be:

- ITACyL weather stations.
- CHE weather stations.
- CHD weather stations.
- MSG data.
- WRF model data.
- Volunteer observer weather stations.

The attributes that characterize an instance of *source* are simply an identifier and a full name, as shown in figure 2.

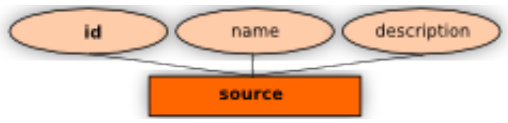

Figure 2: **source entity.**

In the future, more data could be included in the Open Data repository, such as:

- Hail pad network data.
- Radar data.
- lightning detection networks.

Source-place relationship

An instance of *place* has an origin, which means it must have an instance of *source* associated with it. Also, an instance of *source* can be attached to multiple instances of *place*. The relationship between *source* and *place* entities is shown in Figure 3.

Figure 3: **source-place relationship.**

Variable entity

Geographical points can be represented using entity *place* instances, although they are not interesting on their own. The aim is to know the value of certain meteorological variables at a given geographical point. An instance of *variable* entity represents a meteorological variable, such as temperature, relative humidity, wind direction or wind speed. The *variable* entity and its attributes is shown in figure 4.

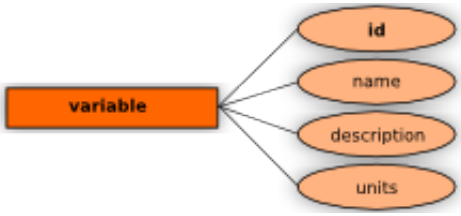

Figure 4: **variable entity.**

Place-variable relationship

Meteorological data can be modelled using *place* and *variable* entities. The relationship between an instance of *place* and an instance of a *variable* at a given time represents a measurement. The relationship between *place* and *variable* entities, and its own attributes, is shown in Figure 5. The *Datetime* attribute represents the given time and *value* attribute represents the variable value for the measurement.

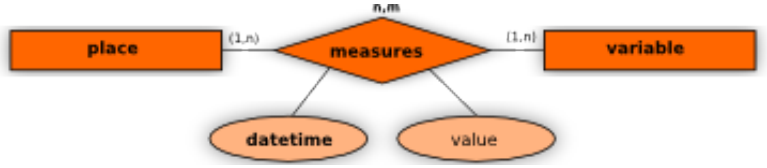

Figure 5: **Place-variable relationship.**

ARCHITECTURE

Once the data model has been defined, it is necessary to focus on the architecture. This should take several aspects into account:

- Input data: It should consider the different formats of the data that will be contained in the Open Data.
- Output data: It should consider the different formats of the data that will be obtained from the Open Data.
- Data repository: A repository has to be chosen that is capable of containing information in accordance with the data model, optimizing all accesses as far as possible.
- Input services: It needs to define a mechanism for entering information in different formats into the Open Data.
- Output services: Similarly, it also needs to define a mechanism for obtaining information in different formats from the Open Data.

Figure 6 shows the Open Data repository architecture.

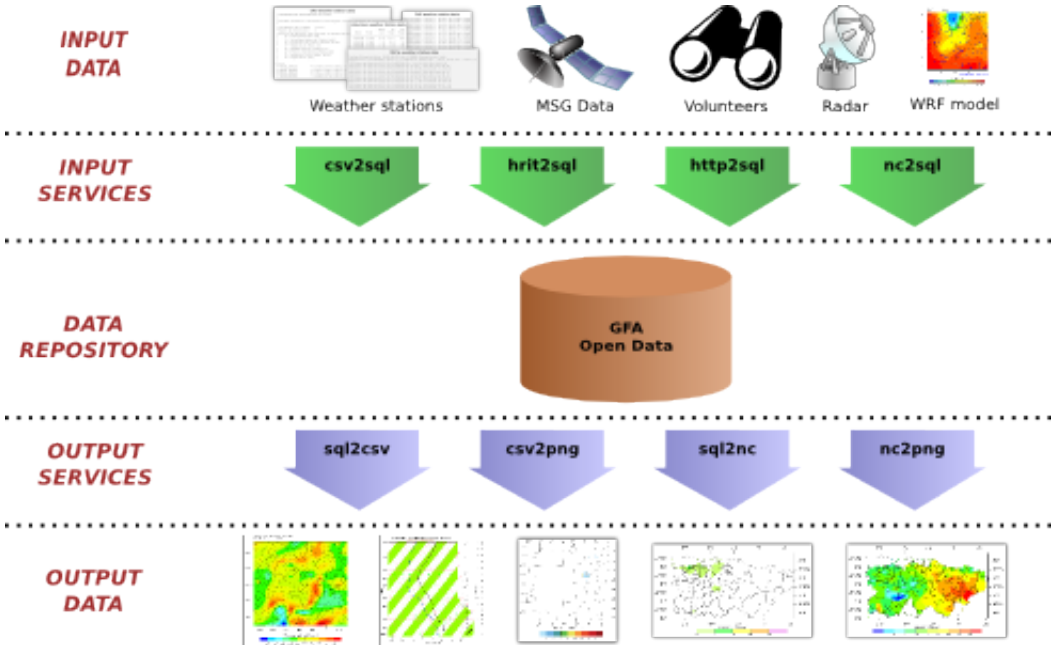

Figure 6: **GFA Open Data repository architecture.**

Data repository

Probably, the most important aspect is deciding on the type of repository. It seems clear that a relational database must be used in order to house the previously discussed data model. The GFA Open Data repository is hosted in a *PostgreSQL* server.

Input data

As previously mentioned, we have to manage different formats, such as:

- *NetCDF*: NWP models typically use this format for their outputs.
- *Text/csv*: Weather stations typically use this format. However, the precise format depends on the manufacturer of the meteorological station.
- *HRIT: MSG data is received using this format.*

At present, the GFA Open Data repository can use *Text/csv* and *NetCDF* formats for entering data.

Input services

Users need to have a mechanism that allows them enter information in the Open Data repository. GFA Open Data will provide four modules for entering data: *csv2sql, nc2sql*, *http2sql* and *hrit2sql*, discussed below.

Csv2sql

This module has been designed to enter *text/csv* data in the relational database. As previously discussed, this is a common input data format, normally used by weather stations, etc.

This module is used to manage weather station data sent by CHD, ITACyL and CHE. Data is sent using the FTP protocol and is managed by the GFA server in order to enter it in the Open Data repository. Figure 7 shows a number of files received by the GFA server with CHD, ITACyL, CHE and volunteer weather stations data.

As can be seen in Figure 7, even though all the files are formatted as text, each has its own format. *Csv2sql* must be ready to manage all of them in order to generate an intermediate comma separated values file, which can be use to enter data into the database. *Csv2sql* is implemented using *Python*. The script can be downloaded from the GFA's website[3].

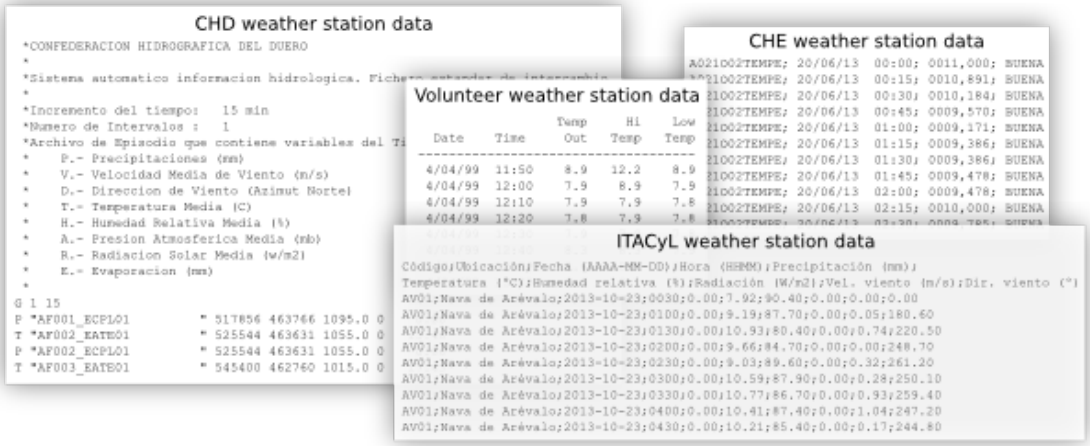

Figure 7: **CHE, CHD, ITACyL and volunteer weather statation data.**

Nc2sql

This module has been designed to enter *NetCDF* data in the relational database. As discussed, most NWP models obtain their outputs using this format. GFA obtains a weather forecast every day using he WRF model. *Nc2sql* is used to enter WRF model outputs into the database. *Nc2sql* is implemented using *Python*. The script can be downloaded from the GFA's website[3].

Http2sql

The GFA is interested in collecting data from volunteer observers. There are a large number of amateur observers who have weather stations and other meteorological instruments, whose measurements can be of great interest to the scientific community. In order to do this, the GFA provides an application[4] for volunteer observers which allows them to send their data to the Open Data repository using the *http* protocol, which means an internet connection is needed.

Http2sql is implemented using PHP, and at present is the only input service that can be used by anyone through the following address: hawking.unileon.es/station/station.php. In order to enter data into the database a number of arguments are required, as shown in Table 1.

Argument	Description
id	Place identifier.
Date	Measurement date.
Time	Measurement time.
day	Measurement day.
month	Measurement month.
year	Measurement year.
hour	Measurement hour.
minute	Measurement minute.

Table 1: **http2sql arguments.**

Some of these arguments are mandatory in order to enter data, or no data is recorded in the database. One mandatory argument is *Id* which identifies a place, such as a weather station. Http2sql verifies that the place is registered in the database as a *place* entity instance. This means that if an observer wants to enter their data in the Open Data repository, they first have to inform the GFA so that their coordinates are registered in the database as a *place* entity instance. Other mandatory arguments have to do with the measurement date and time; the precise time of all measurements must be known.

Once the measurement time and place has been set, a variable and its value are needed. This data is given as a new argument. Each variable has an identifier which corresponds to the argument name. This correspondence is set by the *variable* entity. If the variable is not downloaded into the database, no data is recorded. A number of valid *http2sql* invocations are shown in table 2.

Single variable (temperature) invocation	Two variable (temperature and relative humidity) invocation	Multiple variable invocation
hawking.unileon.es/station/station.php? **id**=obv_003_ben& **day**=24&**month**=7& **year**=2013&**hour**=14& **minute**=00& **Temp_Out**=26.8	hawking.unileon.es/station/station.php? **id**=obv_003_ben& **Date**=24/07/13& **Time**=14:00& **Temp_Out**=26.8& **Out_Hum**=35	hawking.unileon.es/station/station.php? **id**=obv_003_ben& **Date**=24/07/13& **Time**=14:00& **Temp_Out**=26.8& **Out_Hum**=35& **Wind_Speed**=4.8& **Wind_Dir**=S& **Rain**=0.00

Table 2: **http2sql invocations.**

Hrit2sql

This module has been designed to enter HRIT data into the relational database. MSG data is received in this format. At present, HRIT data is converted beforehand to NetCDF. Once the data has been converted to NetCDF, *Nc2sql* is used to enter it into the database.

Output data

The output data that users can obtain, depends directly on the output services provided by the GFA Open Data repository. Users can obtain data as raw data in csv/text or NetCDF format, or as custom graphics displaying meteorological data. At present, registered users can obtain information from the GFA Open Data repository through the GFA website[4].

Output services

Users need to have a mechanism that allows them to obtain information from the Open Data repository. GFA Open Data will provide several modules that can be used to obtain data: *sql2csv, sql2nc*, *csv2png, nc2png*, *csv2kml* and *nc2kml*, as discussed below.

Sql2csv & sql2nc

Sql2csv and s*ql2nc* has been designed to obtain data in text/csv and NetCDF format respectively. Both modules have been implemented using Python. The scripts can be downloaded from the GFA's website[3].

Csv2png, csv2kml, nc2png & nc2kml

Csv2png and c*sv2kml* have been designed to provide graphic and *kml* representations from the data obtained using *sql2csv. Nc2png* and *nc2kml* have been designed to provide graphic and *kml* representations from the data obtained using *sql2nc.* All of them have been implemented using Python. The scripts can be downloaded from the GFA's website[3].

At present, these modules can be invoked using the http protocol either using the data collection form on the GFA's website[2], shown in Figure 8, or by invoking a *create-map* PHP service directly from the following address: hawking.unileon.es/station/create-map.php.

Figure 8: **Data collection form from GFA website.**

In order to obtain some outputs, certain arguments are required by *create-map*. Table 3 shows some of these arguments. A number of valid *create-map* invocations are shown in table 4.

Argument	Description
type	Meteorological variable for plotting
date	Measurement date and time.
out	Output type: - *png* for graphical representations. - *kml* for a representation visible on Google Maps and Google Earth.
region	Predefined Region. At present: - *cyl* for Castile-León. - *nw* for the Duero and Ebro river basins.
maxlat	Maximum latitude for custom region.
maxlon	Maximum longitude for custom region.
minlat	Minimum latitude for custom region.
minlon	Minimum longitude for custom region.
resolution	Grid resolution.

Table 3: **Create-map arguments.**

Getting a *kml* representation for temperature on October 3rd 2013 at 16:00	**Getting a graphic representation for rain on October 3rd 2013 at 16:00**
http://hawking.unileon.es/station/create-map.php? **type**=temperature& **date**=201310031600& **out**=kml	http://hawking.unileon.es/station/create-map.php? **type**=rain& **date**=201310031600& **out**=png

Table 4: **Create-map invocations.**

APPLICATIONS

This section discusses some of the applications developed from the information contained in the GFA Open Data repository.

Verification of the precipitation

"Precipitation field" is a highly irregular variable, which means it is difficult to verify due to the great variability it presents on a small spatial scale. The traditional verification indices only provide incomplete information, as they are based on a point-by-point comparison and not on the underlying spatial information.

In order to verify the precipitation forecast accuracy (Garcia-Ortega et al., 2012), observed precipitation matrices from CHE rain gauges were compared with precipitation forecast matrices from the MM5 model. The matrices of the observed/forecast precipitation fields were produced by rescaling the values of the observed and forecasted precipitation with a resolution of
0.09°, using a Cressman interpolation (Cressman,1959).

Likelihood of hail

Severe storms with hail precipitation are one of the most common meteorological risks in Europe. They can provoke grave economic losses and even affect the population. Identifying these types of episodes is one of the most usual demands from the population in general, and from risk management bodies in particular. The GFA has developed a model that allows for the real-time identification of hailstorms (nowcasting), using information gathered from an MSG (Merino et al., 2013; Guerrero et al, 2013).

A correction has been made to the MSG satellite coordinates in order to correct the Parallax effect, according to Vicente et al. (2002). To make the correction, it is necessary to know the height of the cloud at each point. To do so, the value of the brightness temperature 10.8 μm variable is compared with a vertical temperature profile obtained from the output given by the WRF model.

CONCLUSIONS & FUTURE WORKS

Providing meteorological information in Castile-León as Open Data is a powerful tool that allows researchers to validate and correct their algorithms and affirmations. The previous section shows some examples of how researchers can use this data in their own research. Having such a large amount of information is a major advantage in itself, but if this information is made available to any user without any limits, the advantage is even greater, as researchers do not have to waste time gathering data or contacting different agencies, and can focus on their work. As already mentioned, there are currently no repositories of this kind in Castile-León.

The unification of various data sources into a single repository that is available to all types of users has great potential for creating a wide range of end applications, ranging from risk management bodies such as the emergency services, through to end users.

Nevertheless, there is still a great deal to be done, such as adding new data sources, building an Open Data Portal, or improving and securing input/output services.

ACKNOWLEDGEMENTS

This study was funded by two research projects awared by the Junta de Castilla y León: LE220A11-2, LE003B009.

Additionally, the authors would like to thank the CHD, CHE, ITACyL for the use of data from their weather stations networks.

REFERENCES

Cressman, G.P., (1959) An operational objective analysis system. Mon. Wea. Rev., **87**, pp 367–374.

Garcia-Ortega, E., Merino, A., López, L., Sánchez, J.L., (2013) Role of mesoscale factors at the onset of deep convection on hailstorm days and their relation to the synoptic patterns. Atmospheric Research, **114-115**, pp 91-106.

Guerrero-Higueras, A.M., Merino, A., López, L., Sánchez, J.L., Matellán, V., (2013) Identification of summer hailstorm from MSG data using Python. Third Symposium on Advances in Modeling and Analysis Using Python, AMS annual meeting, Austin, TX, EEUU.

Merino, A., López, L., Sánchez, J.L., García-Ortega, E., Cattani E. and Levizzani V, (2013) Day-time identification of summer hailstorm cells from MSG data. Nat. Hazards Earth Syst. Sci., 13. 2695-2705.www.nat-hazards-earth-syst-sci.net/13/2695/2013/ doi:10.5194/nhess-13-2695-2013

Vicente, G., J. Davenport, and R. Scofield, (2002) The role of orographic and parallax corrections on real time hight resolution satellite rainfall rate distribution. Int. J. Re-mote Sens., **203**, pp 221–330.

1. Atmospheric Physics Group. Universidad de León. Contact: am.guerrero@unileon.es
2. Dpto. Ingenierías Mecánica, Informática y Aeroespacial. Universidad de León.
3. http://gfa.unileon.es/?q=es/toolbox
4. http://gfa.unileon.es/?q=node/143