Development of a REST API for obtaining site-specific historical and near-future weather data in EPW format

Hu Du[1], Phillip Jones[1], Eva Lucas Segarra[2], Carlos Fernández Bandera[2]
[1]Welsh School of Architecture, Cardiff University, Cardiff, United Kingdom
[2]School of Architecture, University of Navarra, Pamplona, Spain
DuH4@cardiff.ac.uk

Abstract

Obtaining site-specific high accuracy historical and near-future weather data have always been a challenging task for building simulation community to do either building performance analysis or predictive building control.

Although 'typical' (such as TRY/TMY) or 'extreme' (such as DSY/DRY/EWF) weather files are made available, they often do not fit the purpose of studies.

This paper demonstrates a novel approach to obtain real-time current and forecast weather in EPW format for building simulation using the free online toolchain. It is the first attempt to create weather API (Application programming interface) designed explicitly for building simulation community.

Introduction

Use actual weather data for building simulation is crucial for studying the weather impact on peak demand. Instead of using TMY, Hong, Chang et al. (2013) have used past 30-years actual weather data to assess long-term building performance and to support energy policy making and energy code development.

Discussions on how to represent historical data have always been a hot topic. Researchers around the world have developed many methods of producing single year weather file. Manuel, Sukumar et al. (2017) provided a comprehensive review of these past and future weather data for building simulation. 'Typical' weather files from nine countries, seven 'extreme' weather files and various weather generator models have been discussed. However, an important element, real-time weather data, has not been addressed.

Due to the availability of real-time sensing data and the popularity of simulation programs, real-time simulations have been becoming feasible in recent years. A framework for simulation-based real-time whole building performance assessment has been developed by Pang, Wetter et al. (2012).

Real-time weather data, whether cloud computing (Chang 2017), and cloud computing building simulation platform (Richman, Zirnhelt et al. 2014) are the essentials elements for real-time simulation and near-future prediction.

Du, Jones et al. (2017) have developed an approach to predict real-time urban heat island effect and indoor overheating at the urban scale. This is based on their work to understand the reliability of near-future weather data for building performance prediction in the UK (Du, Jones et al. 2016). High accuracy weather forecasts are crucial for real-time simulation. Apart from meteorological office's CFD models for the weather forecast, satellite (Pierro, De Felice et al. 2017) and camera (Chu, Zheng et al. 2017) have been used as a weather sensor for predict weather information. However, these methods are still reasonable expensive and need specialised equipment.

Oil, aviation, shipping and fishing industry have been using bespoke designed weather services for many years. Although the cost is gradually decreasing, it is still expensive for individual building occupants. In recent years, few technology-driven companies started to embed forecast models into their commercial products for optimising energy system performance, such as Google Nest Thermostat and Tesla Powerwall. However, their platform and algorithm are not accessible for the public.

In recent years, researchers or practitioners are demanding an online service that covers as many locations as possible, and most importantly provides real-time site specific historical and near-future weather data for their predictive building control and model validation studies.

There are many established weather forecast services providers around the world, and many of them have made their data free available for smartphone APPs access through APIs. Table 1 has listed 8 API weather forecast providers known to the authors, and they allow free access for the public. They cover over 200,000 cities around the world. AEMET is Spanish meteorological agency operating under the Ministry of Agriculture, Food and Environment. The MetOffice is the United Kingdom's national weather service and funded under the Department for Business, Energy and Industrial Strategy. Met.no is part of Norway's national meteorological institute and supported by Ministry of Education and Research. All the rests are commercial companies who offer free access with certain limitations, such as how often you can request and the total number of requests during certain period.

Table 1: Weather forecast API providers

API provider	Forecast	Interval	Format
aemet	Next 7-day	hourly	JSON
metoffice	Next 5-day	hourly	XML
met.no	Next 10-day	3-hourly	XML
openweathermap	Next 5-day	3-hourly	XML/ JSON
weatherbit	Next 5-day	3-hourly	JSON
darksky	Next 7-day	hourly	JSON
wunderground	Next 10-day	hourly	XML/ JSON
apixu	Next 10-day	hourly	XML/ JSON

Their forecasts are based on global and regional scale CFD models for atmosphere. They could cover the atmosphere with a grid size of 1-25km horizontally and up to 70 layers vertically for about 40km high.

Method

This study aims to test technical feasibility of developing a REST API for users to obtain site-specific historical and near-future weather data. The feasibility testing was conducted in Postman environment which uses the JavaScript language for pre and post request process.

Instead of developing the API from scratch, a right place to start with API development is connecting into other APIs first, in this case, above weather forecast APIs listed in table 1. The response from these APIs could be past 24-hour historical weather data and next 5-10 days hourly/3-hourly weather forecast. A brief comparison between parameters in EPW weather file and parameters in JSON/XML file is shown in table 2.

Table 2: Weather parameters used in EnergyPlus simulation and presence in JSON forecast

No.	EPW	E+	JSON
1	Year	Y	Y
2	Month	Y	Y
3	Day	Y	Y
4	Hour	Y	Y
5	Minute	Y	Y
6	Weather code	N	Y
7	Dry bulb temperature	Y	Y
8	Dew point temperature	Y	Y/A
9	Relative humidity	Y	Y
10	Pressure	Y	
11	Extraterrestrial horizontal radiation	N	
12	Extraterrestrial direct normal radiation	N	
13	Horizontal infrared radiation from Sky	Y	Y/A
14	Global horizontal radiation	Y	Y/A
15	Direct normal radiation	Y	Y/A
16	Diffuse horizontal radiation	N	
17	Global horizontal illuminance	N	
18	Direct normal illuminance	N	
19	Diffuse horizontal illuminance	N	
20	Zenith luminance	N	
21	Wind direction	Y	Y
22	Wind speed	Y	Y
23	Total sky cover	Y	Y/A
24	Opaque sky cover	N	
25	Visibility	N	Y/A
26	Ceiling height	N	
27	Present weather observation		
28	Present weather codes		
29	Precipitation water	N	
30	Aerosol optical depth	N	
31	Snow depth		
32	Days since last snowfall	N	
33	Albedo	N	
34	Liquid Precipitation Depth		
35	Liquid Precipitation Quantity	N	
Y: yes, N: no, Y/A: yes with additional work			

As shown in the table, parameters used by EnergyPlus simulation (column 3) are available in API forecast JSON file. Key parameters such as temperature, relative humidity, wind speed and wind direction are included in the API forecast response. However, some parameters, such as dew point temperature, radiations, need further work to be calculated using known parameters. Depending on the forecast providers, API forecast responses also provide some additional information, such as UV index, a text description of the weather, a satellite image of cloud which could help calculate missing parameters.

API is an interface that allows two software programs to communicate with each another over Internet uses HTTP requests to transfer data. REST (REpresentational State Transfer) is a communications approach that is often used in the development of Web services. REST does not use much bandwidth, which makes it a better fit for use over the Internet.

The REST API offers standard HTTP methods (e.g., OPTIONS, GET, PUT, POST, and DELETE) for exchange information. The GET method is the most common function to obtain information which is identifiable by the request URL. For requesting the weather data, the request URL includes the location of the place interested and a key which helps identify the client (the person/machine making the request). The location could be latitude and longitude, city name or city ID number. Example of request URL can be found in step 1 of the following JavaScript code.

```
Step 1:

GET
https://opendata.aemet.es/opendata/api/prediccion/especifica/municipio/horaria/31201/?api_key={{AEMET_api_key}code}

{
   "descripcion": "éxito",
   "estado": 200,
   "datos":
"https://opendata.aemet.es/opendata/sh/d320c34f",
   "metadatos":
"https://opendata.aemet.es/opendata/sh/93a7c63d"
}

Step 2:
```

```
tests['Status Code is 200'] =
(responseCode.code===200);

if (responseCode.code === 200) {
      try {
      var body_text=JSON.parse(responseBody),

            datosURL=body_text.datos,

            metadatosURL=body_text.metadatos;
   }
   catch(e) {
      console.log(e);
   }

postman.setGlobalVariable("datosURL",datosURL);
postman.setGlobalVariable("metadatosURL",metadat
osURL);
}

Step 3:

GET {{datosURL}}
```

Figure 2: JavaScript code in Postman

The above JavaScript demonstrated the whole process of obtaining next 7-day hourly forecast for a Spanish city in Postman environment. Step 1: Sending the request including location ID and API key. Step 2: Obtaining one of the URL from the response received from step 1. Step 3: Request the next 7-day hourly weather forecast by obtaining information from the above URL. The final response and Postman working environment are illustrated in figure 3. It demonstrated the technical feasibility of developing a REST API for users to obtain site-specific historical and near-future weather data.

Postman is the essential toolchain for API developers to share, test, document and monitor APIs. More than 3 million engineers and developers worldwide use Postman to build connected software via APIs.

Authors do not intend to develop API for building simulation as part of this study, because companies like Autodesk and JEPlus have been working on this area.

APIs have been used for analysis and assessment of smart city architecture (Badii, Bellini et al. 2017). Bus and train timetable APIs are the most commonly used ones for the general public. In building simulation community, building simulation API, such as GBS Web Service API, enables any third party application to access the building energy simulation service. However, there is no specialised weather service for building simulation community.

For post data processing, simplified prediction models can be developed using machine leading APIs, such Google Prediction API. Users can submit training data of

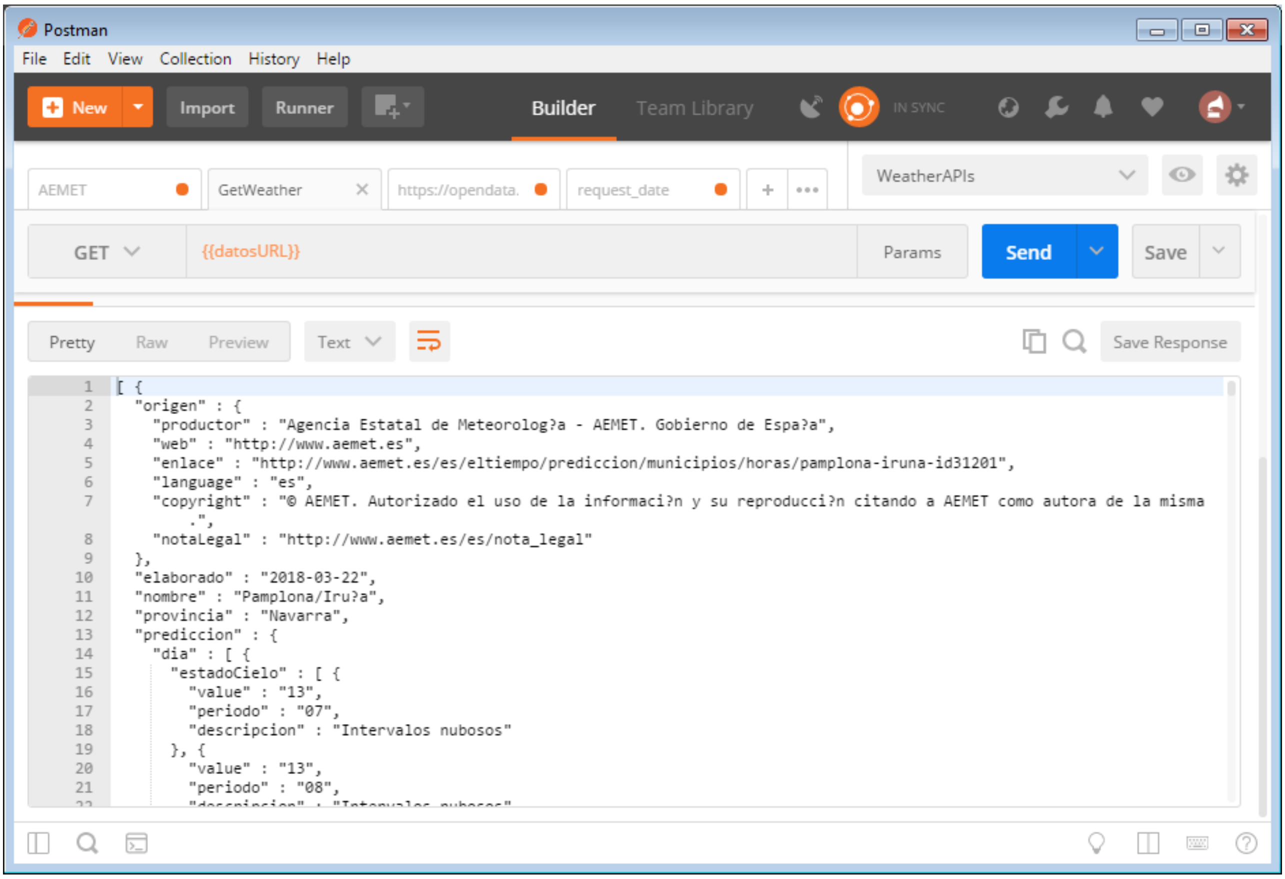

Figure 3 Postman testing environment and forecast response

indoor, outdoor temperature and energy demand, to create machine learning models to predict indoor temperature and energy demand based on newly submitted outdoor temperature.

Figure 4 summarised the workflow of above method.

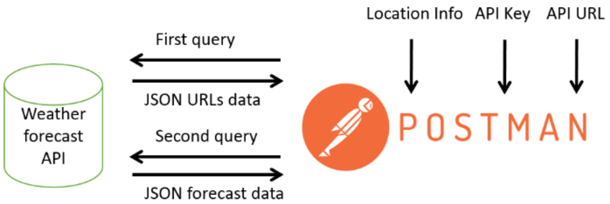

Figure 4: Illustration of workflow

Results

To avoid introducing further uncertain factors during building simulation, this study only exams the difference between forecasts and observations using the data for city of Pamplona, Spain.

The comparisons between forecast and observation were conducted using a local weather station (Observation 1) installed at University of Navarra, a nearby weather station belonging to Spain National Weather Meteorological Agency (Observation 2) and forecasts from four weather APIs which offer hourly forecasts. Two sets of observations help understand the implications due to equipment accuracy, installation, and surrounding environment of measurements. This provides a benchmark for comparing the difference between forecasts and observations.

Figure 5 plotted temperature from two observations sources and four forecasts sources during the period of 6 February to 15 March 2018 (38 days).

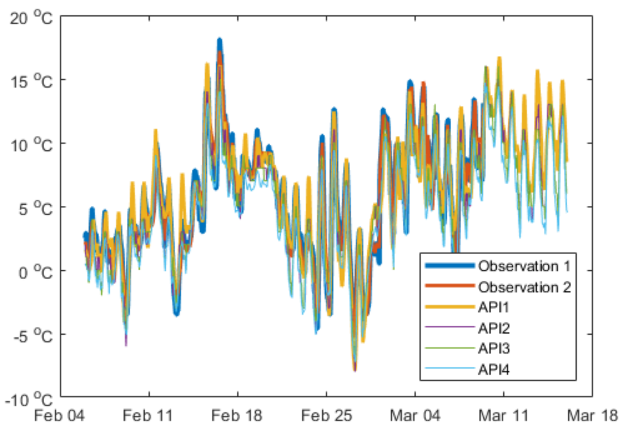

Figure 5: Temperature observations and forecasts (6 Feb-15 Mar 2018)

In general, the forecasts are very close to observation data. In order to quantify the difference, the Root Mean Square Error (RMSE) between Observation 1 and Observation 2, forecasts from four APIs. The RMSE compares differences between two sets of data and the formula for calculating RMSE is shown in equation (1). The example size (n) of this study is 912 data points covering hourly data for 38 days.

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n}(x_{1,t}-x_{2,t})^2}{n}} \quad (1)$$

The result in figure 6 shows that the temperature differences between forecasts and Observation 1 is in range of 1.6 °C to 2.2 °C depending on which providers made the forecasts. The temperature difference between two sets of observations is 0.7 °C. Therefore, the conclusion can be made that using forecasted data instead of observed data only introduced less than 1 °C error. Comparing with the accuracy of common temperature sensors (0.2-0.5 °C), the error is acceptable for general building energy modelling exercises.

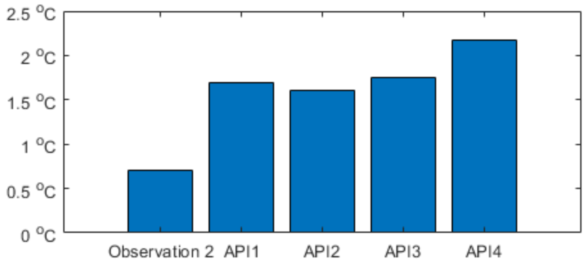

Figure 6: The Temperature RMSE between Observations and forecasts (6 Feb-15 Mar 2018)

The similar conclusion can also be drawn for relative humidity. Figure 7 shows that the relative humidity difference between two sets of observations is 7%, whereas the differences between forecasts and Observation 1 is in range of 10% to 13% depending on the providers. Note that the common used relative humidity sensors have the accuracy of 2.5-3.5%. This is roughly equivalent to the difference between different forecasts providers. Therefore, the selection of weather providers has limited impact on the forecast accuracy of relative humidity.

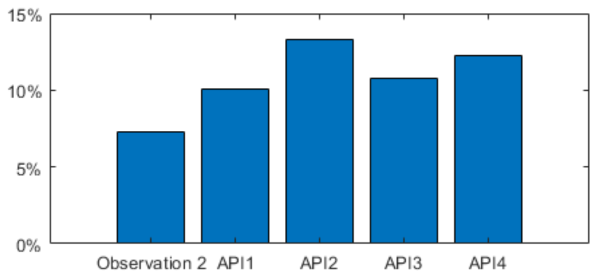

Figure 7: The RH RMSE between Observations and forecasts (6 Feb-15 Mar 2018)

In general, weather forecast providers often make next 5-10 days' forecast available to the public. They can be in the hourly or 3-hourly interval. The longer time span of forecasts, the better preparation can be made in the building management systems. Energy storage system, particularly thermal storage, and renewable systems need more than one day ahead forecasts to provide the optimal system performance.

Tests have been made to exam the errors between the observation and next day forecasts, two-day ahead forecast, three-day ahead forecast, and up to seven-day ahead forecast. Figure 8 shows next 48-hour forecast is the best, and 3-4 day ahead forecast is reasonably accurate. The accuracy decreases with the time moving further forward.

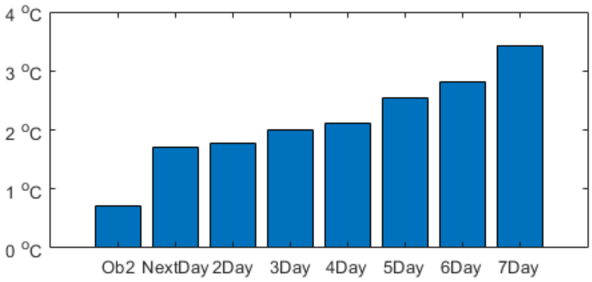

Figure 8: The Temperature RMSE between Observations and 7-day ahead forecasts (6 Feb-15 Mar 2018)

Conclusion

With the increasing number of weather APIs available for smartphone Apps, building simulation community would also obtain weather data by requesting the past 24-hour observations and next 7-day forecasts from weather API servers periodically and use for building simulation.

This paper demonstrates a novel approach to obtain real-time current and forecast weather in EPW format for building simulation using the free online toolchain. It is the first attempt to create weather API specifically designed for building simulation community. Users and machines can directly request real-time weather files for predictive building control, model calibration and real-time building performance analysis/benchmarking.

The comparisons between forecast and observations show that next 48-hour forecasts from good weather forecast providers can be used to replace data from observation stations due to the high accuracy of forecasts. This created rich datasets for studying energy storage optimisation.

The work enables the machine to machine data communication so that automatic building simulation and real-time optimisation became feasible. It also reduces the effects of installing weather stations and processing large sets of meteorological data. Due to the high accuracy of the forecast, it could potentially reduce the error introduced by weather file selection (up to 30%).

Future research

Obtaining real-time weather data is the essential element of real-time building simulation and predictive control. This is often achieved through machines to machines communication. JSON and XML are the typical formats of web applications. With the future trend of cloud building energy simulation application, the form of weather data should be re-considered by the building simulation community. The commonly used simulation programs, such as EnergyPlus, IES, and DesignBuilder should start considering to support XML or JSON weather file as an input. Dedicated weather forecast API for building simulation community should also be established to promote free access to historical and near-future weather data.

Acknowledgement

This study is funded by the Welsh government's Sêr Cymru (Stars Wales) program. The authors thankfully acknowledge weather forecast data provided by Spanish Met Office, api.openweathermap.org, api.weatherbit.io, api.darksky.net, api.wunderground.com, and api.apixu.com and api.met.no.

References

Badii, C., P. Bellini, D. Cenni, A. Difino, P. Nesi and M. Paolucci (2017). "Analysis and assessment of a knowledge based smart city architecture providing service APIs." Future Generation Computer Systems 75: 14-29.

Chang, V. (2017). "Towards data analysis for weather cloud computing." Knowledge-Based Systems 127: 29-45.

Chu, W.-T., X.-Y. Zheng and D.-S. Ding (2017). "Camera as weather sensor: Estimating weather information from single images." Journal of Visual Communication and Image Representation 46: 233-249.

Du, H., P. Jones and C. Long (2017). A novel approach to predict real-time urban heat island effect and indoor overheating. Passive Low Energy Architecture (PLEA) 2017.

Du, H., P. Jones and B. Ng (2016). Understanding the reliability of localized near future weather data for building performance prediction in the UK. 2016 IEEE International Smart Cities Conference (ISC2).

Hong, T., W.-K. Chang and H.-W. Lin (2013). "A fresh look at weather impact on peak electricity demand and energy use of buildings using 30-year actual weather data." Applied Energy 111: 333-350.

Manuel, H., N. Sukumar, A. C. David, K. Tristan, P. R.-G. Alfonso, E. Matthew, F. Daniel and W. Michael (2017). "A review of current and future weather data for building simulation." Building Services Engineering Research and Technology 38(5): 602-627.

Pang, X., M. Wetter, P. Bhattacharya and P. Haves (2012). "A framework for simulation-based real-time whole building performance assessment." Building and Environment 54: 100-108.

Pierro, M., M. De Felice, E. Maggioni, D. Moser, A. Perotto, F. Spada and C. Cornaro (2017). "Data-driven upscaling methods for regional photovoltaic power estimation and forecast using satellite and numerical weather prediction data." Solar Energy 158: 1026-1038.

Richman, R., H. Zirnhelt and S. Fix (2014). "Large-scale building simulation using cloud computing for estimating lifecycle energy consumption." Canadian Journal of Civil Engineering 41(3): 252-262.