2018 Building Performance Analysis Conference and
SimBuild co-organized by ASHRAE and IBPSA-USA
Chicago, IL
September 26-28, 2018

AN OPEN SCIENCE APPROACH FOR BUILDING PERFORMANCE STUDIES

Steven K Firth[1], Gareth Cole[2], Tom Kane[1], Farid Fouchal[1] and Tarek M Hassan[1]

[1] School of Architecture, Building and Civil Engineering, Loughborough University, UK
[2] Loughborough University Library, Loughborough University, UK

ABSTRACT

Open Science is increasingly viewed as a priority for collaboration and advancement of scientific disciplines. This paper presents two Open Science methods for building performance studies: i) a method for storing building performance information using an open data format, including the creation of a custom XML schema and the novel use of both XML and CSV formats for data storage; and ii) a method for carrying out building performance analysis in an open and reproducible manner using the Jupyter Notebook tool.

The work is based on the open-access REFIT Smart Home dataset (publically available at https://doi.org/10.17028/rd.lboro.2070091.v1), a published dataset of building performance information in 20 UK homes. The dataset includes detailed building survey information and over 1.3 billion sensor readings.

The paper concludes with a discussion on the future of Open Science approaches for building performance studies, and how the processes described in the paper can be applied to both measurement and simulation datasets.

INTRODUCTION

Kraker et al. (2011) in their paper *'The case for an open science in technology enhanced learning'* propose a vision for Open Science based on four instruments:

1. Open Access - publishing the results of academic research as freely available on the public internet
2. Open Data - publishing the datasets collected in the research process, without restricting their use
3. Open Source - making software developed in the research available under an open license
4. Open Methodology - sharing the methodology of a study, and the tools used for data collection and analysis

These combine to create an Open Science approach where each stage of the research process is made openly available to the wider research community. This ambition is recognised at the highest levels of scientific thinking, such as the first recommendation of the *'Science as an Open Enterprise'* report by the UK's Royal Society:

> 'Scientists should communicate the data they collect and the models they create, to allow free and open access, and in ways that are intelligible, assessable and usable for other specialists in the same or linked fields wherever they are in the world. Where data justify it, scientists should make them available in an appropriate data repository. Where possible, communication with a wider public audience should be made a priority, and particularly so in areas where openness is in the public interest.'
>
> (Royal Society, 2012)

The FAIR Guiding Priniciples, as described in Wilkinson et al. (2016), provide guidance as how to achieve an Open Science approach. FAIR stands for **F**indable, **A**ccessible, **I**nteroperable and **R**eusable and there is a specific emphasis on 'enhancing the ability of machines to automatically find and use the data'. The FAIR principles are not solely intended for data but 'also to the algorithms, tools and workflows that led to that data'.

This paper studies the Open Science approach and its application to building performance studies. Here building performance studies refer to either studies of existing buildings or simulation studies of new, hypothetical buildings. There are good examples of developments in Open Science in the building performance community, such as the software-neutral interoperable gbXML data format (gbXML, 2018). The journal Nature Scientific Data also provides examples of best practice and current developments in the area (Nature Scientific Data, 2018).

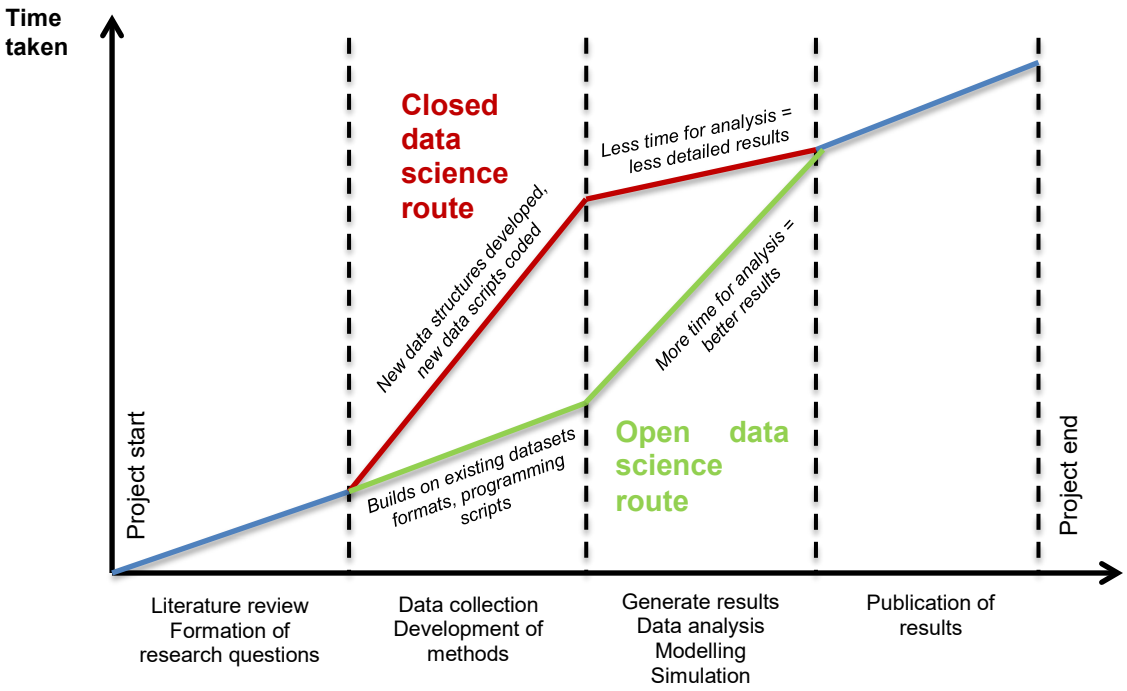

Figure 1: Time taken for Open and 'Closed' data-science approaches for a typical building performance research project

BENEFITS OF OPEN SCIENCE

Open Science improves the speed and quality of scientific advances within a discipline, and there are many examples of the benefits that an Open Science approach can bring to researchers.

More citations

Open-access data can lead to increased visibility and citation of published research. Piwowar and Vision (2013), in their paper titled *'Data reuse and the open data citation advantage'*, found 9% increase in citation rates for a study of 10,555 publications which had data available in a public repository. McKiernan et al. (2016) in their paper *'How open science helps researchers succeed'* also discuss the advantages beyond citation rates, demonstrating how 'open research is associated with increase in citations, media attention, potential collaborators, job opportunities and funding opportunities'.

Avoiding unnecessary data collection and methods duplication

Figure 1 illustrates the advantages of working within a community which has embraced Open Science. For a given research project, much less time will be needed to collect data and develop analysis methods, as researchers can easily build on those of previous studies. This provides much more time for the analysis and generation of research results, leading to better publications.

Less errors

The Open Science route in Figure 1 will also potentially reduce errors in the research. Data analysis techniques could be developed from previous work, reducing potential errors and the time required. Once results are generated these could be more easily checked with previous studies, improving the robustness of the results and validation of the research methodology.

Furthering Open Science

Perhaps most importantly of all, active researchers who are using previous research developed with Open Science instruments will have a blueprint for how to work in an Open Science manner. This makes it much easier for the current research to also employ these open science instruments and for the new research to become an open, accessible and valuable addition to the body of knowledge.

AN OPEN SCIENCE WORKFLOW FOR BUILDING PERFORMANCE STUDIES

Can such a vision of Open Science be a reality for the Building Simulation community? Open Access publications occur regularly in the building simulation field, with many journal papers being released in open-access form. Open Source software has been successfully demonstrated through the success of EnergyPlus which is available to use under an open license and also releases the original source code for advanced users (EnergyPlus, 2018). This has helped make EnergyPlus one of the most widely used building simulation software in the world.

However Open Data and Open Methodology approaches are less well developed, with many studies not releasing either the data collected or the methods used to collect and analyse the data. This is in part because this is a difficult task to do and requires forethought and planning throughout the research process. It may also be because releasing the underlying data and methods of research is not generally required

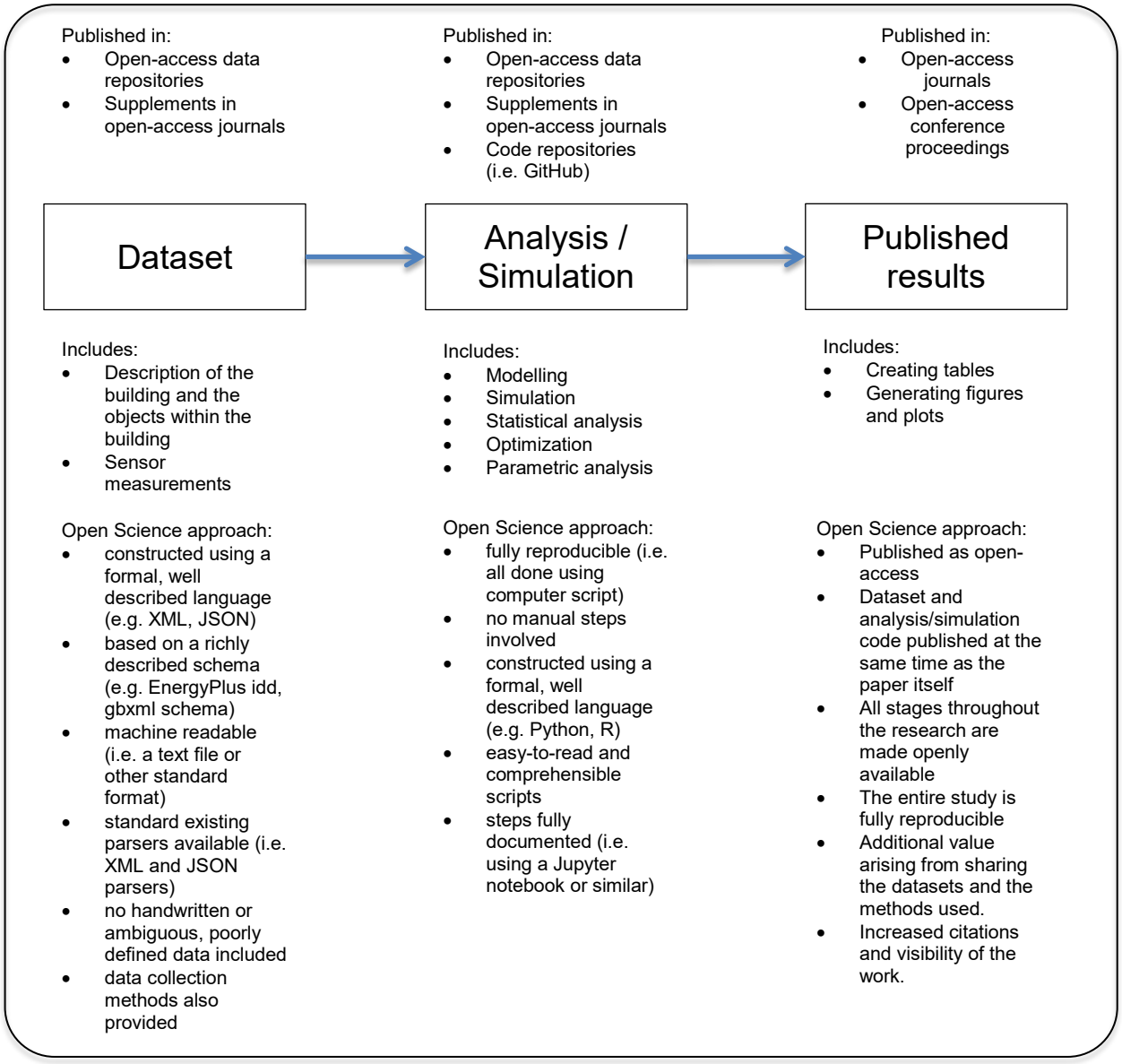

Figure 2: An Open-Science Workflow for the Building Simulation Community and suggested actions to meet the FAIR (Findable, Accessible, Interoperable and Reusable) principles

by funding bodies at present or generally expected by the Building Simulation community.

Figure 2 shows an Open Science workflow for a typical building performance study. It would be possible for most published work in the Building Simultion field to follow this workflow and provide Open Data and Open Methodology information for the wider researh community. The ambition here is that a published journal paper would not only be published as open-access, but the underlying dataset is also published as well as the computer scripts used to carry out the analysis on the dataset. The entire study could be published so that all stages are openly available and the study is completely reprodicible.

Dataset

The underlying dataset could represent either an existing building or the design of a hypothetical new building. The dataset should be completely in computer-readable format, i.e. a computer file. Typical files might include model input files, such as an EnergyPlus idf file, or a Building Information Model (BIM) files such as gbXML or IFC. A schema file describing the structure of the dataset should be specified, such as an EnergyPlus idd file or a gbXML schema file. Industry standard methods to represent building assets and performance are available, such as the Green Building XML format (gbXML, 2018). However further advancements are needed to overcome the limitations of such formats, such as recording changes to the building parameters over time. The methods used to create the dataset (such as the building survey undertaken) should be documented.

Analysis / Simulation

Analysis and simulation includes data anlaysis of the raw underslying dataset and using the dataset as an input to simulation tools. This process should be publishable and entirely reproducible. For this reason the complete process should be carried out in well-documented computer scripts, which can then easily be shared with and understood by other. Computer scripts can be used to carry out all types of data analysis and used to automate simluation runs. This stage of the process may well be the most useful to future researchers, as well-documented computer code would be a valuable resource for future research in a similar area.

Published results

The final results of the study can be formally published as a peer-reviewed journal or conference paper. At this stage more computer scripts might be used to create the final tables and graphs for the published paper, to avoid manual steps which greatly reduce the reproducibility of the work. The underlying datasets and analysis code are published at the same time as the paper itself, either as supplementary material in the journal itself or in an external data repository. The entire research study is both reproducible and reusable.

CASE STUDY: THE REFIT SMART HOME DATASET

Using a case study example of 20 UK homes, this section demonstrates an Open Data and Open Methodology approach specifically suited to building performance studies.

Overview of the REFIT study

In 2012, 20 homes were recruited in the East Midlands area of the UK to take part in a building performance study. This was part of a wider research project interested in the Smart Home concept and energy savings. The project was a UK Research Council funded project named REFIT and ran from 2012 to 2015 (REFIT, 2017). The data collected from the 20 homes forms the REFIT Smart Home Dataset which has been published as an Open Data dataset (to access the dataset follow the link in Firth et al., 2017). Information about the characteristics of the homes and the people in the homes was collected through site visits and over 100 sensors were placed in each home to capture aspects of the building performance. Figure 2 shows a selection of the temperature and gas sensors used. Other sensors included electricity meter sensors, appliance plug sensors, motion detectors and Smart Home sensors. The sensors were placed during the building survey visits and recorded measurements in the homes for 18-24 months.

Final dataset

The final REFIT Smart Home dataset has been published as an open-access dataset (Firth et al., 2017) and is stored on the Loughborough University Data Repository. At the time of writing, the dataset had been cited in one journal publication (Kane et al., 2017), had been downloaded 98 times and viewed 788 times. The REFIT project also collected additional datasets which have been made available by the partner universities, including over 1.5 billion power measurements from the electricity meters and appliances in the homes (Murray et al., 2017).

Building Survey Data (XML)

The building survey data of the REFIT Smart Home Dataset is stored in a refitXML file (a custom XML schema developed in this work). The 'REFIT_BUILDING_SURVEY.xml' file contains this data and an excerpt for one of the homes, 'Building01,

is shown in Figure 3. Reading directly from the xml file, it can be seen that Building01 is a detached home, facing at 327° from North and has a cavity wall construction. Building01 contains a Space (e.g. a room) which is a heated study with a floor area of 6.25m^2. Within the study room a Hobo pendent sensor ('Sensor41') was placed which recorded the air temperature ('TimeSeriesVariable41') at 15 minute intervals from 2nd October 2013 to 3rd December 2013. This nesting of objects, sensors and variables proves to be very useful in working with the dataset.

Sensor Measurements Data (CSV)

Sensor measurements could be stored in the refitXML file but the size of the measurement data makes this impractical. Very large xml files can be difficult to parse by external readers, and the use of tags in the xml format results in a larger file size than other formats. Instead the sensor data is stored using a comma-separated variable (csv) file. An excerpt of the 'REFIT_TIME_SERIES_VALUES.csv' is shown in Table 1. The 'TimeSeriesVariable/@id' column is an identifier column and associates the sensor readings with a TimeSeriesVariable node in the xml file. The 'dateTime' column provides a timestamp for the sensor readings using the standard xml date and time format. The 'data' column shows the value of the sensor reading itself. In this way, Table 1 shows that on the 2nd of October 2013 at 05:00 variable TimeSeriesVariable1 had a value of 17.772. To understand what this value represents, the user must look up the information about TimeSeriesVariable1 in the separate xml file (in this case the 17.772 value is an air temperature reading by a Hobo U12 sensor recorded in a bathroom in Building01).

Table 1: An excerpt of the REFIT_TIME_SERIES_VALUES.csv file showing the first 8 rows of data

TimeSeriesVariable/@id	dateTime	Data
TimeSeriesVariable1	2013-10-02T05:00:00Z	17.772
TimeSeriesVariable1	2013-10-02T05:30:00Z	18.081
TimeSeriesVariable1	2013-10-02T06:00:00Z	18.176
TimeSeriesVariable1	2013-10-02T06:30:00Z	18.176
TimeSeriesVariable1	2013-10-02T07:00:00Z	18.105
TimeSeriesVariable1	2013-10-02T07:30:00Z	18.01
TimeSeriesVariable1	2013-10-02T08:00:00Z	17.891
TimeSeriesVariable1	2013-10-02T08:30:00Z	17.772
…	…	…

```
<Building id="Building01" startDateTime="2013-10-01T00:00:00Z"
   occupancyType="Single family dwelling" builtFormType="Detached house or bungalow"
   orientation="327" wallTypeMainBuilding="Masonry-Boxwall-Cavity"
   wallAgeBandMainBuilding="1975 - 1980" cavityWallInsulationPresent="Yes"
   windowType="Double glazed - UPVC" loftType="Fully boarded"
   loftInsulationType="Mineral wool/fibre glass" loftInsulationThickness="300mm">
   ...
   <Space id="Space1" startDateTime="2013-10-01T00:00:00Z" conditionType="Heated"
      area="6.25" volume="14.375" storeyLevel="0" roomType="Study">
      <Sensor id="Sensor41" startDateTime="2013-10-02T05:00:00Z" endDateTime="2013-
         12-03T15:15:00Z" manufacturer="Onset" model="Hobo pendant">
         <TimeSeriesVariable id="TimeSeriesVariable41" startDateTime="2013-10-
            02T05:00:00Z" endDateTime="2013-12-03T15:15:00Z" variableType="Air
            temperature" units="C" intervalType="FixedInterval" intervalUnit="Minute"
            intervalLength="15" hasMissingData="No" repeatsOmitted="No"
            hasDuplicateTimestamps="No"/>
      </Sensor>
      ...
   </Space>
   ...
</Building>
```

Figure 3: An excerpt of the REFIT_BUILDING_SURVEY.xml file which describes an actual building 'Building01' and the properties of and relationships between Building01, Space1, Sensor41 and TimeSeriesVariable41.

table_example

This notebook shows a method to create tables based on the REFIT Smart Home Dataset.

In [52]:
```
from lxml import etree; from collections import Counter; import pandas as pd
from IPython.display import display, HTML
```

In [53]:
```
def attribute_text(element_list,name,keys=None):
    "Returns a string for the table with counts for unique values in attribute 'name'"
    c=Counter(b.get(name) for b in element_list)
    if not keys: keys=c.keys()
    return '; '.join(['{} ({})'.format(k,c[k]) for k in keys])
```

In [54]:
```
def floor_area_list(tree, buildings):
    "Returns a list of total heated floor areas for each building"
    for b in buildings:
        heated_spaces=b.xpath('./a:Space[@conditionType="Heated"]',namespaces=NS)  # a list of heated Space elements
        a=sum([float(s.get('area')) for s in heated_spaces])                       # the total floor area for all heated Spaces
        area.append(a)
    return area
```

In [55]:
```
xml=r'REFIT_BUILDING_SURVEY.xml'                                    # the name of the refitXML file
NS={'a':'http://www.refitsmarthomes.org'}                           # the xml namespace of the refitXML file
tree=etree.parse(xml)                                               # parses the refitXML file
buildings=tree.getroot().xpath('//a:Building',namespaces=NS)        # finds all the Building elements in the xml file
area=floor_area_list(tree, buildings)                               # a list of total heated floor areas
ages=['1850 - 1899','1919 - 1944','1945 - 1964','1965 - 1974','1975 - 1980',
   '1981 - 1990','1991 - 1995','Post 2002']
l=[]                                                                # sets up a list for the table columns
l.append(('House type',attribute_text(buildings,'builtFormType')))
l.append(('Construction type',attribute_text(buildings,'wallTypeMainBuilding')))
l.append(('Construction age',attribute_text(buildings,'wallAgeBandMainBuilding',keys=ages)))
l.append(('Floor area','Mean: {:0.1f} m2, Range: {:0.1f} - {:0.1f} m2'.
          format(sum(area)/len(area),min(area),max(area))))
l=list(zip(*l))
df=pd.DataFrame({'Characteristic':l[0],'Description':l[1]})         # a pandas DataFrame of the final table
with pd.option_context('display.max_colwidth', -1):                 # unlimited printed column width
    display(HTML(df.to_html(index=False)))                          # display table
```

Characteristic	Description
House type	Detached house or bungalow (16); Semi detached house or bungalow (3); Mid terrace house or bungalow (1)
Construction type	Masonry-Boxwall-Cavity (15); Masonry-Boxwall -Solid (5)
Construction age	1850 - 1899 (2); 1919 - 1944 (3); 1945 - 1964 (2); 1965 - 1974 (6); 1975 - 1980 (1); 1981 - 1990 (3); 1991 - 1995 (1); Post 2002 (2)
Floor area	Mean: 126.1 m2, Range: 73.4 – 229.2 m2

Figure 4: Browser screenshot of a Jupyter notebook which creates a publication-ready table summarising the characteristics of the buildings in the REFIT Smart Home Dataset.

Jupyter Notebooks

This section presents an example of an Open Methodology workflow for the data analysis stage. Through a review of data analysis software, in this work the choice was made to develop the analysis code in the Python programming language, as Python was specifically designed to be easy-to-read and share (Python, 2018). The Python analysis code is shared within Jupyter notebooks, which can hold the code itself, the outputs of the code and additional written documentation (Project Jupyter, 2018). The use of notebooks for documenting and sharing analysis code is a method that has gained much interest within the research community (for example, see Shen, 2014).

Figure 4 shows a screenshot of a Jupyter notebook used to calculate the first 4 rows of a results table. Notebooks appear and are edited within an internet browser (linked to a local server) and so the figure shows a screenshot of the browser itself with the top and bottom sections cropped. The notebook shows five 'cells'. The upper cell is a text cell and in this example shows a title and basic description about the notebook. Text in these cells is written in 'markdown', a simple tag-based language which can provide formatting such as headers, bold, bullet points etc. The lower four cell contains computer code, written in Python, in the grey shaded boxes. The final cell also contains the output of the code once it has been run, in the area below the grey shaded box at the bottom of the figure.

The output of the notebook in Figure 4 shows the house type, construction type, construction age and floor area of the 20 homes. However, and more importantly in this paper, it shows the exact method of how these results were calculated. The Python code imports the xml and csv files of the REFIT dataset directly and carries out the analysis tasks to provide these statistics. This is a complete process, and the steps taken from the original

data to the final results can be exactly traced by other users.

DISCUSSION

Limitations of the approaches presented

This paper has presented solutions for the Open Data and Open Methodology instruments in the context of Building Performance studies. The methods given here are not intended as final or generic solutions, but rather represent a first step and illustrative approaches which others may adapt for future studies.

The Open Data approach based on refitXML represents a solution which was specifically developed for the data collected in the REFIT project. It represents a significant improvement on simply releasing the raw data as collected, but a number of limitations still exist:

1. In the published dataset itself the original data collection instruments (such as the blank paper building surveys etc.) have not been provided. This was partly intentional, as the questions themselves and the response options are documented in the refitXML schema file so the original data collection instruments may not be required. However some users may prefer to have the actual questions, and the format and structure of the questions within the questionnaire.
2. Secondly some important information is not provided in the public version of the dataset due to confidentiality issues. In particular the 3D geometry of the buildings was not made publically available as it was deemed that this could potentially be used to directly identify the individual households. This data is retained in an internal version of the dataset but is stripped out of the public release with only the orientation and areas of surfaces made available.
3. The use of the xml format has proved challenging for some users. Computer scientists and programmers have found the structure logical and simple to use, but other users more familiar with the table structures of Excel or relational databases have initially struggled to interpret the dataset. It could be argued that the choice of using xml is failing the criteria that the Open Data should be structured for ease-of-use. However in this case, for a study which captures not only building level information but information on a multitude of objects within the building itself, structuring the dataset in database-style tables does not simply the structure. Rather the equivalent database contains 30+ tables with many primary-foreign key links which results in long and complex SQL query expressions.
4. There are also often privacy and security issues that prevent the sharing of building performance datasets. These need to be considered at the start of the data collection process and appropriate consent forms from participants need to be completed.

The main limitation of the Open Methodology approach for the data analysis is the use of the Python programming language which will only be familiar to a subset of users. However if the conclusion is accepted that programming code is the only way to create truly documented and reproducible data analysis, then this is an intrinsic limitation.

The Future of Open Data Science for Building Performance Studies

The solutions presented in this paper have shown that it is possible to use Open Data and Open Methodology approaches in Building Performance studies. These solutions are intended as illustrative and not final, generic approaches for all cases.

One vision for the implementation of these approaches is the publication of journal papers which both release the data used in an Open Data fashion and release the data analysis methods used as an Open Methodology. A further step could be to publish an entire journal paper as a Jupyter notebook, where the text, table and figures of the paper are provided, and the programming code to create the results is also present. This would create a single document in which the results of a paper and the methods used to create the results are combined in a single file, and enable other researchers to study and modify the data analysis methods for their own studies.

The solutions in this paper could equally be used for building simulation studies which employ modelling as the main method for results generation. Input files to models (such as EnergyPlus idf files) can be made open access and the data analysis steps used to analyse the model output files (such as EnergyPlus csv results files) could be provided as Jupyter notebooks. In a parametric analysis, the code used to modify model input files could also be made open.

The ease of reuse of data analysis techniques and programming code is largely impacted on the data structures under analysis. One could imagine a series of Python functions and classes for analysing EnergyPlus results files and, because this is a clearly defined and widely used format, these function and classes could be easily used by others. For building survey data, BIM formats such as IFC and gbxml may provide a common format which could be widely used. However these

approaches would need to be further developed for widespread use, or new approaches may be required such as the refitXML format as described in this paper. Improving the data structures used within Building Performance studies and Building Simulation could be a useful goal for the Building Simulation community.

CONCLUSION

This paper has described the collection of building survey and sensor data for 20 homes in the REFIT project. The paper describes the benefits of using an Open Science approach for the publication of this dataset and in the analysis of the data. An Open Data approach using refitXML is described for developing an open, structured and documented dataset from raw data which has been made publically available for other researchers to use. An Open Methodology approach is also described which creates an open, reproducible and documented method for data analysis using Python and Jupyter notebooks.

The key conclusions are:

- It is possible to create building performance datasets based on Open Data principles. However the existing data formats are not suited to all studies and new format may need to be developed, such as the refitXML format as described in this paper.
- Python and Jupyter notebooks provide a solution to publishing data analysis techniques and meeting Open Methodology requirements.

Further work includes further development of data structures and formats for Building Performance studies and Building Simulation studies. More development of the data analysis techniques is also planned, with the goal of publishing an academic journal paper in which the data analysis methods are fully documented and published using the Jupyter notebook approach.

ACKNOWLEDGMENT

This work has been carried out as part of the REFIT project ('Personalised Retrofit Decision Support Tools for UK Homes using Smart Home Technology', Grant Reference EP/K002457/1). REFIT is a consortium of three universities - Loughborough, Strathclyde and East Anglia - and ten industry stakeholders funded by the Engineering and Physical Sciences Research Council (EPSRC) under the Transforming Energy Demand in Buildings through Digital Innovation (BuildTEDDI) funding programme. For more information see: www.epsrc.ac.uk and www.refitsmarthomes.org.

REFERENCES

EnergyPlus. 2018. https://energyplus.net/

Firth S.K., Kane T., Dimitriou V., Hassan T.M., Fouchal F., Coleman M. and Webb L. 2017. *REFIT Smart Home dataset*, figshare, https://dx.doi.org/10.17028/rd.lboro.2070091

gbXML (2018). *gbXML: an industry supported schema for sharing building information between disparate building design software tools*, available at http://www.gbxml.org/

Kane T., Firth S.K., Hassan T.M. and Dimitriou V. 2017. *Heating behaviour in English homes: An assessment of indirect calculation methods*, Energy and Buildings, 148, pp.89-105 http://dx.doi.org/10.1016/j.enbuild.2017.04.059

Kraker P., Leony D., Reinhardt W. and Beham G. 2011. The case for an open science in technology enhanced learning, Int. J. Technology Enhanced Learning, Vol. 3, No. 6, pp 643-654

Nature Scientific Data, 2018. A peer-reviewed, open-access journal for descriptions of scientifically valuable datasets, https://www.nature.com/sdata/

McKiernan et al. 2016. *How open science helps researchers succeed*, eLife 2016;5:e16800. DOI: 10.7554/eLife.16800

Murray D., Stankovic L., and Stankovic V. 2017. *An electrical load measurements dataset of United Kingdom households from a two-year longitudinal study*, Scientific Data, vol. 4, Article number: 160122. http://dx.doi.org/10.1038/sdata.2016.122

Piwowar and Vision. 2013. *Data reuse and the open data citation advantage*. PeerJ 1:e175; DOI 10.7717/peerj.175

Project Jupyter. 2018. http://jupyter.org/

Python. 2018. https://www.python.org/

REFIT. 2017. *The REFIT project website*, www.refitsmarthomes.org

Royal Society. 2012. *Science as an open enterprise,*

The Royal Society Science Policy Centre report 02/12, ISBN: 978-0-85403-962-3.

Shen, 2014. Interactive notebooks: Sharing the code, Nature News, www.nature.com/news/interactive-notebooks-sharing-the-code-1.16261

Wilkinson, M. D. et al. 2016. *The FAIR Guiding Principles for scientific data management and stewardship.* Sci. Data 3:160018 doi: 10.1038/sdata.2016.18 (2016).