TECHNISCHER BERICHT / TECHNICAL REPORT

Daten Aufbereitung / Data Preparation

für das EnOB-Verbundvorhaben

BUiLD.DIGITiZED

IoT und BIM für die Inbetriebnahme und den Betrieb
von netzdienlichen Niedrigstenergiegebäuden

1.7.2020 – 30.6.2023

MONITORING.digital
FKZ: 03EN1021A

Hochschule Offenburg | Institut für Energiesystemtechnik
Badstraße 24 | 77652 Offenburg

INBETRIEBNAHME.digital
FKZ: 03EN1021B

Fraunhofer-Institut für Solare Energiesysteme ISE
Heidenhofstraße 2 | 79110 Freiburg

EVALUATION.digital
FKZ: 03EN1021D

Mondas GmbH
Emmy-Noether-Str. 2 | 79110 Freiburg

Luftqualität
FKZ: 03EN1021F

Testo SE & Co. KGaA
Celsiusstraße 2 | 79822 Titisee-Neustadt

Anlagenplanung mit BIM
FKZ: 03EN1021C

Konzmann Gebäudetechnik GmbH
Niederwiesenstraße 34 | 78050 VS-Villingen

Betriebsführung mit BIM
FKZ: 03EN1021E

Maurer Energie- und Ingenieurleistungen GmbH & Co. KG
Dr.-Kurt-Steim-Straße 7 | 78713 Schramberg-Sulgen

Autor dieses Berichtes:

B.Sc. Ishan Chopra / M.Sc. Danny Carvajal

Hochschule Offenburg | Institut für Energiesystemtechnik
Badstraße 24 | 77652 Offenburg

Offenburg, 23. Februar 2023

Abstract:

This report provides comprehensive literature on the data preparation for simulation and operation models. The Input data for simulation models must be prepared and fit defined criteria for which a variety of data extraction and transformation methods must be applied. This is discussed profoundly in the report regarding the data preparation approaches taken on the raw data from various sensors for the project. Section 1 discusses about the intro to the process of data preparation. Section 2 studies the approaches towards data preparation tasks and delve into working of different Python scripts. Section 3 briefly focuses on the after process of data preparation, shows structural chart of operation models and results from simulation.

1. Introduction:

In general, the essential task of data preparation comprises of four main sub tasks:

1. Data Pre-processing
2. Data extraction
3. Data Imputation
4. Data Consolidation

Firstly, according to the project's scenario, the main sub tasks must be divided among specific set of processes which are carried out to get the final data in the required format. Table 1 shows an of data preparation tasks.

Table 1 *Overview of data preparation tasks*

General data preparation sub-tasks	Project's Data preparation tasks
1. Data Pre-processing	• IoT Web Platform: Mondas
2. Data extraction	Gather: • Raw data from Mondas • Weather data from Hochschule Offenburg Weather Station (until the new Pyranometer of the RIZ Energie starts its operation)
3. Data Imputation	Gap filling and classification
4. Data Consolidation	Merge data into desired format

The process of data preparation has to be executed monthly to get the result as a "Messdaten Excel file" also termed as "Combined Excel File" in the report. The "Messdaten Excel file" has the raw sensor data from Mondas in the desired format and is further required to successfully run the operation model simulation also called digital twin of the RIZ Energie.

In this report, it is discussed how different methods progressed during the project to tackle these the sub-tasks of data preparation. This report will mainly focus on methods of Data extraction, Data Imputation and Data Consolidation.

2. *Methods:*

Moving forward, methods will be discussed to the above-mentioned sub tasks in Table 1. Initially, Method 1 was used for monthly data preparation which proceeded to Method 2. The advantage of method 1 is that it is still used during troubleshooting of data from a specific sensor and for a different time period.

Starting from the scopes of both methods and then a detailed explanation for different tasks is given below. These are procedural steps which also are performed in the same way as for the process of data preparation.

2.1. Scopes

Method 1: Manual method

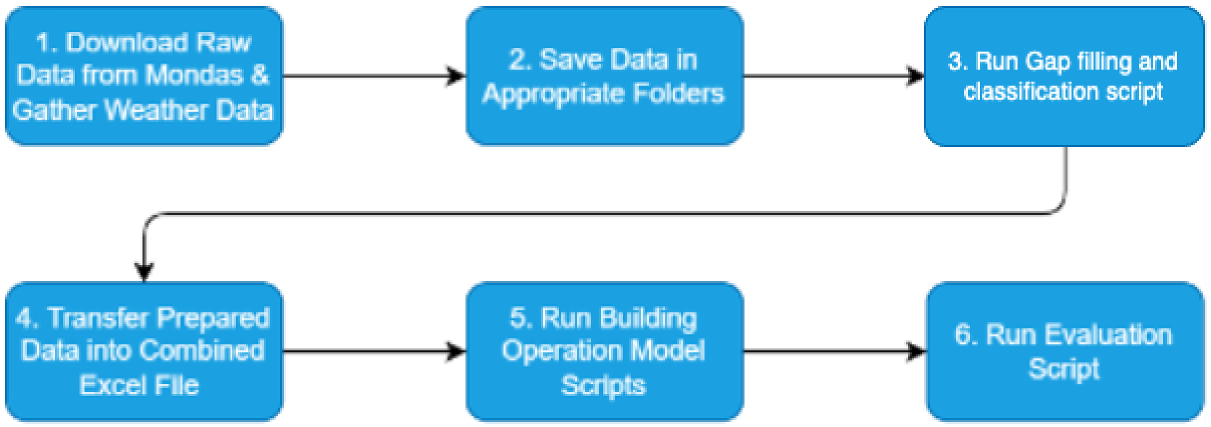

1. Download Raw Data from Mondas & Gather Weather Data
2. Save Data in Appropriate Folders
3. Run Gap filling and classification script
4. Transfer Prepared Data into Combined Excel File
5. Run Building Operation Model Scripts
6. Run Evaluation Script

Figure 1 *Initial process steps of monthly data preparation*

During the initial phase of the project, the process of data preparation as shown in Figure 1, was defined accordingly to ensure the successful run of "Gap_Filling_Classification.py" script which filled and classified all the gaps in the raw data. This step was crucial during the initial phase of the project for the process of data preparation. The majority of other tasks were carried out manually. The steps 5 and 6 are carried out after the data consolidation and are briefly explained in section 3 of this report.

Method 2: Automated method

The flowchart shown in Figure 2 describes the procedure and main components of "Data_Preparation.py" script which call different scripts and functions to complete different tasks. The goal was to automate the remaining sub - tasks of data preparation method 1 using Python scripts. The purpose of the automation is to save time and improve the accuracy in the process of data preparation. For the automation through Python, different Python libraries were used such as Pandas, NumPy, Selenium, Openpyxl etc.

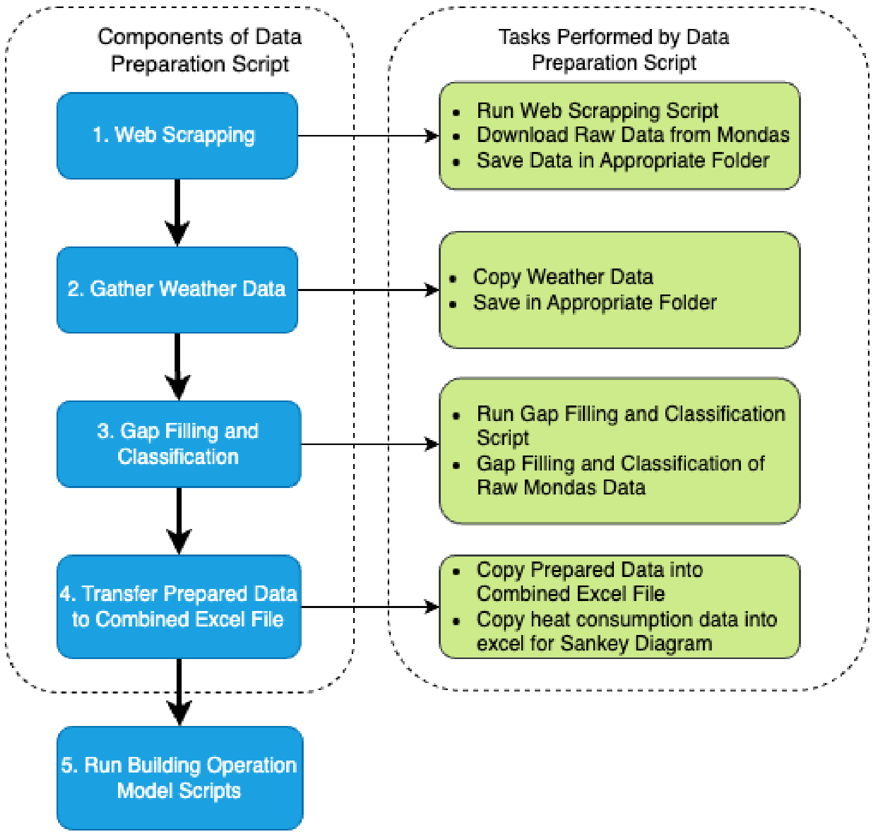

Components of Data Preparation Script
1. Web Scrapping
2. Gather Weather Data
3. Gap Filling and Classification
4. Transfer Prepared Data to Combined Excel File
5. Run Building Operation Model Scripts
Tasks Performed by Data Preparation Script
Run Web Scrapping Script
Download Raw Data from Mondas
Save Data in Appropriate Folder
Copy Weather Data
Save in Appropriate Folder
Run Gap Filling and Classification Script
Gap Filling and Classification of Raw Mondas Data
Copy Prepared Data into Combined Excel File
Copy heat consumption data into excel for Sankey Diagram

Figure 2 *Current process steps of monthly data preparation*

2.2. Data extraction

2.2.1. Download Raw Data from Mondas

This process step is performed to collect all necessary raw sensor data from Mondas. Mondas is web based IoT platform which has all the necessary data from installed sensors. Mondas allows one to create different outputs using the raw data inputs from various sensors.

Method 1:

Initially, the raw sensor data has to be downloaded online from below mentioned link to Mondas Data Export or alternatively the mentioned menu path can be used.

Link to Mondas Data Export:
build-digitized.Mondas.io/Forschung_build_digitized/simulations/eingangsdaten

Alternatively, the following menu path in *Mondas* can be used to get there:
Main navigation → *Simulationsdaten* → *Data Export*

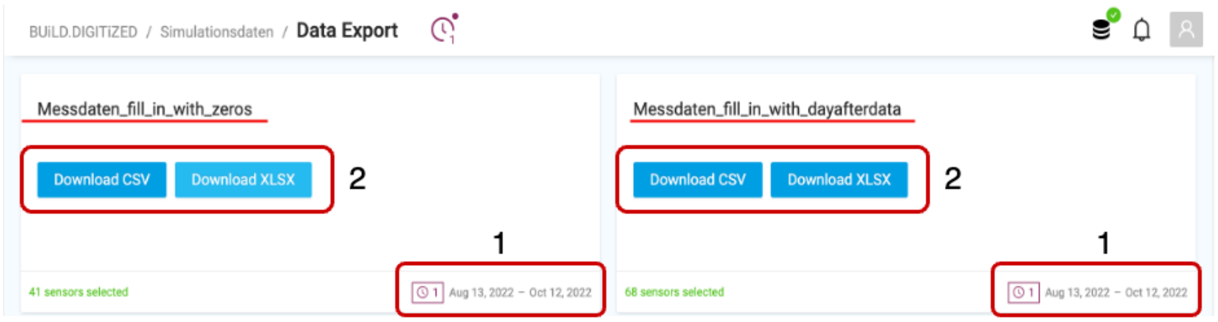

BUiLD.DIGITiZED / Simulationsdaten / Data Export
Messdaten_fill_in_with_zeros
Download CSV
Download XLSX
2
1
41 sensors selected
1 Aug 13, 2022 – Oct 12, 2022
Messdaten_fill_in_with_dayafterdata
Download CSV
Download XLSX
2
1
68 sensors selected
1 Aug 13, 2022 – Oct 12, 2022

Figure 3 *Screenshot of Mondas Data Export webpage*

As shown in Figure 3, there are two widgets on this page at the very top from which all the necessary data can be downloaded. Before downloading, the appropriate time span has to be set up within the widget and then download is performed by clicking on the *Download XLSX*/Download CSV button.

Method 2:

The "Data_Preparation.py" calls the *"Webscrapping_Mondas.py"* and performs the necessary tasks to download the data from two widgets of **Messdaten_fill_in_with_zeros** and **Messdaten_fill_in_with_dayafterdata** from Mondas Data Export. The script downloads the data for a time span of 2 months for each run to prevent any error due to selection of shorter time span even when script has been run later in the month.

The selenium library in Python is used to write the "Webscrapping_Mondas.py". Chrome browser is chosen to automate the process of downloading the data from Mondas because of its wide user base. The download directory of a download has to be defined during the start of the script for each download. That's why "Webscrapping_Mondas.py" runs twice to download all the necessary data from **Messdaten_fill_in_with_zeros** and **Messdaten_fill_in_with_dayafterdata.**

The "Data_Preparation.py" and *"Webscrapping_Mondas.py"* can be found here:
INES\INES\E2G\BUiLD_DIGITiZED\d_Forschung\04_Simulation_Modeling_BIM\02_Python\DigitalTwin_RIZ_15min\DataPreparation_15min

Caution:
During the manual download, to avoid any error/loss of data during the run of "Gap_Filling_Classification.py". It is advised to extend the time range during the time span selection. This could happen because of Python's forward and backward fill methods which essentially requires the very first (e.g.: 01.01.2022 00:00:00) or very last (e.g.: 31.03.2022 23:45:00) row of the needed data. If that is a gap and therefore not present, these gaps at the start or the end will remain. Because of this, the time span of the widget in Mondas has to be extended.

Should one of the widgets not be functioning properly, or just a few selected sensors are of interest, the data can be downloaded separately for each sensor from the same page in Mondas. The downloading procedure is the same as for the two aforementioned widgets. The involved widgets/sensors are given in the Table 2 of Appendix which are differentiated based on gap filling techniques. More is discussed about this in section 2.3.

2.2.2. Weather data

The weather data is an essential information which is required in the operational model simulation. Hence, data must be included in the process of data preparation. The monthly data from weather station of Hochschule Offenburg must be transferred from the below mentioned path to the "*Weatherdata.xlsx*".

Important variables required from weather station of Hochschule Offenburg:

⇒ Global Radiation

⇒ Diffuse Radiation

Once the new pyranometer is set up at the RIZ building, this step will not be necessary anymore. Then these variables will be available at Mondas.

Method 1:

At the beginning, the data had to be copied each month after it's availability in the network folder of weather data to the "*Weatherdata.xlsx*".

Method 2:

Now, the "Data_Preparation.py" performs the task of transferring the measured weather data of one previous month from weather station of Hochschule Offenburg and pastes/merge it an excel "*Weatherdata.xlsx*" that consists of the weather data from the start of 2022. During the transfer, if the data already exists in the excel file, the script outputs "Weather Data already exists" to avoid duplication of same data.

The location of monthly measured weather data from the Weather Station Data at Hochschule Offenburg:
\INES\INES\Wetterdaten\

The location of excel file - "*Weatherdata.xlsx*" excel can be found here:
\INES\INES\E2G\BUiLD_DIGITiZED\d_Forschung\04_Simulation_Modeling_BIM\02_Python\DigitalTwin_RIZ_15min\DataPreparation_15min\HSO_Weather_Data

2.2.3. Save the downloaded data

In order to run the "Gap_Filling_Classification.py" without errors, the data downloaded from *Mondas,* and weather data has to be saved appropriately.

Method 1:

Therefore, the following steps have to be executed:

1) Create three new, empty folders in the same directory where the "Gap_Filling_Classification.py" script is saved with the following titles:
    a. "Mondas_Data_Fill_In_with_Day_After_Data"
    b. "Mondas_Data_Fill_In_with_Zeros"
    c. "HSO_Weather_Data"
2) Paste or save the excel files gathered in sections 1.1 and 1.2 in the appropriate folder.

Method 2:

The downloaded data from Mondas after running the "Webscrapping_Mondas.py" is automatically saved and renamed as the names of Mondas widget and saved at below mentioned folders:

a. "Mondas_Data_Fill_In_with_Day_After_Data"
b. "Mondas_Data_Fill_In_with_Zeros"

The copied weather data is also automatically pasted to "*Weatherdata.xlsx"* and saved to "HSO_Weather_Data".

2.3. Data Imputation

Gap Filling and Gap Classification

The task of data imputation is carried out with help of "Gap_Filling_Classification.py". The script has been mostly the same from **Method 1**. It iterates over the excel files which were saved in the three folders (see step 2.2.3), detects the gaps, fills the gaps, and classify the gaps as (0: No gap, 1: Gap ("0"-value put by us), 2: Gap (Missing value, MONDAS failure)). As shown in Figure 4, it then translates the sensor names to the corresponding column header names from the combined data excel file ("*{date}*_Messdaten_RIZ.xlsx") for each file. On the weather data it also executes a resample in order to obtain 15-minute timestamps.

Techniques for filling the gaps is forward fill, backward fill (if no data is available from the after days) and fill in with zeros. More about gap filling techniques based on different sensor data is explained in the section 5.

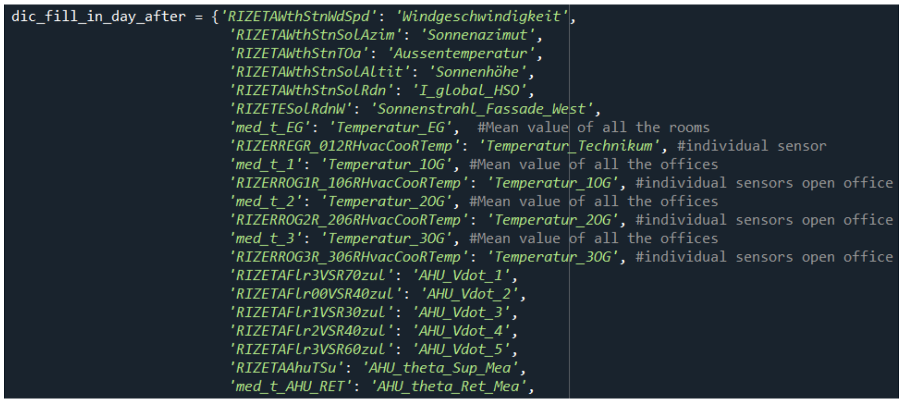
The script then creates new files – if not yet existent in the directory – of the readily prepared data. These files are then saved in a new folder which is named "Prepared_Data" and also created by the script, if not yet existent in the directory. Now all prepared data can be found in this folder.

The "Gap_Filling_Classification.py" can be found here:
INES\INES\E2G\BUiLD_DIGITiZED\d_Forschung\04_Simulation_Modeling_BIM\02_Python\DigitalTwin_RIZ_15min\DataPreparation_15min

Method 2:

Here, the script "Data_Preparation.py" calls the "Gap_Filling_Classification.py" script. The "Gap_Filling_Classification.py" script iterates over the excel files which were saved in the three folders (from section 2.2.3) and creates new files in the "Prepared_Data" folder. Which can be found in the same directory as "Gap_Filling_Classification.py".

Caution:
In case a "TypeError" is occurring when running the script in *Python*, there is a good chance that the reason for the error is a modified sensor name in one of the excel files. After a sensor has been modified in *Mondas*, it sometimes gets assigned a new name. The script uses a dictionary to translate the sensor names (= key in *Python*) to the corresponding column header names (= value in *Python*) from the combined excel data file. If the keys in the dictionary are not completely coincident with the sensor names in the excel files, *Python* will raise an error and stop running the script at the first deviation it encounters.



```
dic_fill_in_day_after = {'RIZETAWthStnWdSpd': 'Windgeschwindigkeit',
                         'RIZETAWthStnSolAzim': 'Sonnenazimut',
                         'RIZETAWthStnTOa': 'Aussentemperatur',
                         'RIZETAWthStnSolAltit': 'Sonnenhöhe',
                         'RIZETAWthStnSolRdn': 'I_global_HSO',
                         'RIZETESolRdnW': 'Sonnenstrahl_Fassade_West',
                         'med_t_EG': 'Temperatur_EG',  #Mean value of all the rooms
                         'RIZERREGR_012RHvacCooRTemp': 'Temperatur_Technikum', #individual sensor
                         'med_t_1': 'Temperatur_1OG', #Mean value of all the offices
                         'RIZERROG1R_106RHvacCooRTemp': 'Temperatur_1OG', #individual sensors open office
                         'med_t_2': 'Temperatur_2OG', #Mean value of all the offices
                         'RIZERROG2R_206RHvacCooRTemp': 'Temperatur_2OG', #individual sensors open office
                         'med_t_3': 'Temperatur_3OG', #Mean value of all the offices
                         'RIZERROG3R_306RHvacCooRTemp': 'Temperatur_3OG', #individual sensors open office
                         'RIZETAFlr3VSR70zul': 'AHU_Vdot_1',
                         'RIZETAFlr00VSR40zul': 'AHU_Vdot_2',
                         'RIZETAFlr1VSR30zul': 'AHU_Vdot_3',
                         'RIZETAFlr2VSR40zul': 'AHU_Vdot_4',
                         'RIZETAFlr3VSR60zul': 'AHU_Vdot_5',
                         'RIZETAAhuTSu': 'AHU_theta_Sup_Mea',
                         'med_t_AHU_RET': 'AHU_theta_Ret_Mea',
```

Figure 4 *Exemplary cutout of a "key: value" dictionary in the data preparation script*

2.4. Data Consolidation

Transfer Prepared Data into Combined Excel File

Method 1:

Here the data prepared in section 2.3, has to be copied and pasted into the combined data excel file ("*{date}*_Messdaten_RIZ.xlsx") while following two rules:

1) Pasting the data into the column with the corresponding column header
2) Pasting the data into the rows with the corresponding timestamps

The combined excel file can be found here:
INES\INES\E2G\BUiLD_DIGITiZED\d_Forschung\04_Simulation_Modeling_BIM\02_Python\DigitalTwin_RIZ_15min\Messdaten_RIZ

Method 2:

This step explains about automating the task of transferring the data from the prepared excel files in folder "Prepared_Data" to combined data excel file ("*{date}*_Messdaten_RIZ.xlsx"). Below given is the logic and explanation of the code in "Data_Preparation.py":

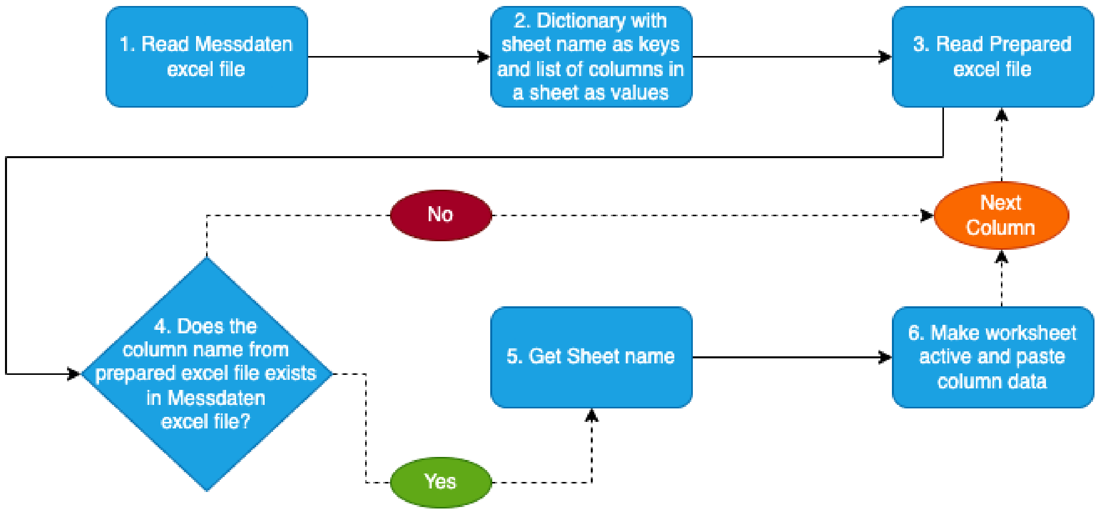

1. Read Messdaten excel file
2. Dictionary with sheet name as keys and list of columns in a sheet as values
3. Read Prepared excel file
No
Next Column
4. Does the column name from prepared excel file exists in Messdaten excel file?
5. Get Sheet name
6. Make worksheet active and paste column data
Yes

Figure 5 *Code's logic for transferring data from prepared to Combined excel*

1) Reading Combined Excel File(“*{date}*_Messdaten_RIZ.xlsx”)

The script reads the (“*{date}*_Messdaten_RIZ.xlsx”) with the help of openpyxl library. Reading an excel file using openpyxl automatically makes one sheet of excel as active worksheet at which further actions can be done. The excel sheets of interest in combined data excel file are *Eingangsdaten, Jalousien, Ausgabedaten, Stromzaehler, WMZ* and *BKT_Ventile*. As the script iterates over the first row of each above-mentioned sheet, a dictionary is created which has keys as the name of sheet and values as list of column names in a sheet. During the iteration through the first row of sheet, if *“timestamp”* is found as column’s name then the timeseries with frequency of 15 mins for the month is pasted after the last filled row of the sheet.

2) Reading Prepared Excel Files and Copying to (“*{date}*_Messdaten_RIZ.xlsx”)

For each prepared excel file, the script reads a file through pandas and subsets based on Datetime index of the data frame for the required month. Using the dictionary created in the previous step, the script checks if the column in a prepared excel file is present in the combined data excel file. If the column is available, script looks for the sheet name among the dictionary keys, makes it the active sheet and then pastes the data after the last filled row of the column in the right excel sheet. This process goes on until all the columns from different prepared excel files are called up.

3) Check for Empty Columns

At the end, a check is done for any empty columns in the combined data excel file and prints the names of it. When there are no empty columns, then the final iteration of combined data excel file (“*{date}*_Messdaten_RIZ.xlsx”) can be found in the “Prepared_Data” folder.

If there are empty column and no adequate data is available for the empty columns. Then paste zero in the columns. Else, data must be downloaded manually for selected sensors from Mondas Data Export after which the necessary process steps of Method 1 must be followed to fill the gaps and transfer the prepared data to the combined data excel file.

3. Results

Run Operation Model Scripts

The data reading and operation model scripts can be found here:
\\Fs3-4-ines\ines\INES\INES\E2G\BUiLD_DIGITiZED\d_Forschung\04_Simulation_Modeling_BIM\02_Python\DigitalTwin_RIZ_15min\Operation_Model_15_Minuten

In this directory, the first script to mention is the data reading script (*Messdatenauslesung_RIZ.py*), which reads all the data from the combined data excel file saved in the directory. The simulation model scripts are interconnected as there is a specific data flow between them (see Figure 6).

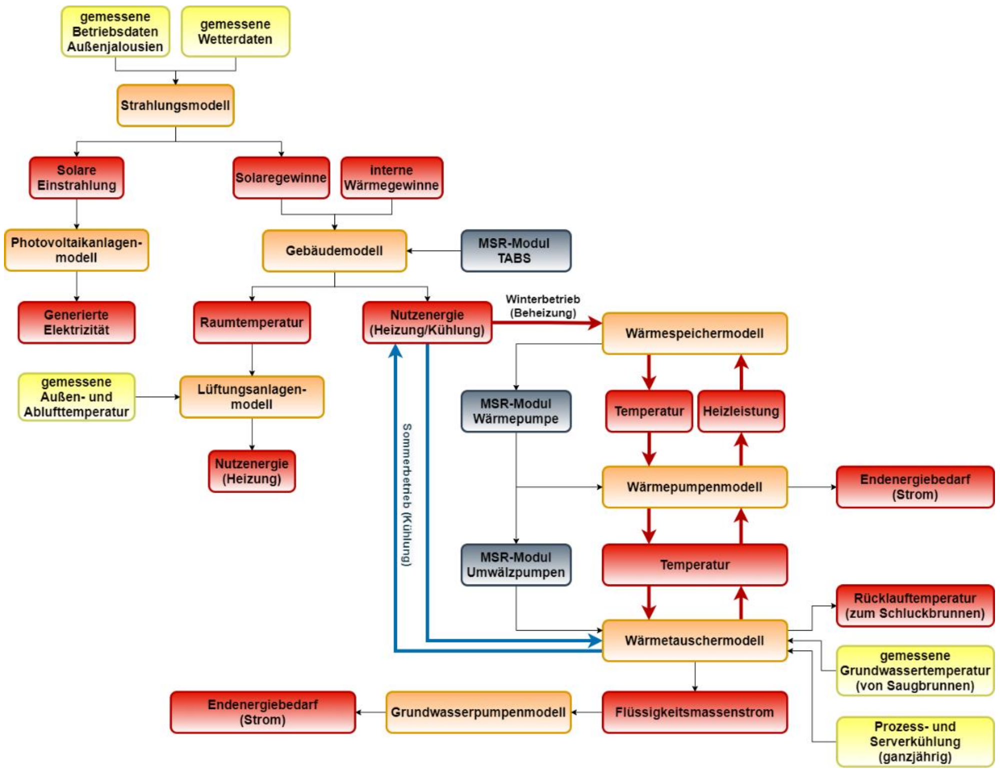


Figure 6 *Data flow diagram simulation models*

However, there is no requirement to run the scripts consecutively in order to be able to run a certain script. The core simulation model script for example, the building model (*Operation_Gebaeudemodell_AlleZonen.py*), can be run directly.

4. Appendix

Below table categorizes different measurements based on the right gap filling technique. This is crucial to decide upon saving the right sensor data in the appropriate folder before the step of data imputation.

Table 2 *Categorization of sensor data based on gap filling technique.*

Data where gaps are filled with zeros	**Data where gaps are filled with day before/after values**
Büros (Vor-/Rücklauftemperatur)	Wetterdaten
Technikum (Vor-/Rücklauftemperatur)	nach Doppelfilter / vor Eingang Schluckbrunnen (Vor-/Rücklauftemperatur)
Prozesskühlung (Vor-/Rücklauftemperatur)	Jalousien
Serverräume (Vor-/Rücklauftemperatur)	Wirkleistung: PV, Wärmepumpe, Brunnenpumpe, Lüftungsanlage (Elektrozähler)
Heizregister (Vor-/Rücklauftemperatur)	Raumtemperatur EG
Wärmetauscher 1 (Serverräume)	Raumtemperaturen Großraumbüros
Wärmetauscher 2 (Prozesskühlung)	Raumtemperatur Technikum
WMZ (Energie)	AHU (Ablufttemperatur)
WMZ (Volumenströme)	AHU (Vdot und Zulufttemperatur)

Discussing about why for few measurements following gap filling techniques are selected. The distinction between the types of gap filling is made because of the different types of measurements:

1) When there are gaps in the raw data of water temperature measurements, they are filled with zeros. When the ground water pumps, or the recirculating pumps are not working then the measurements of temperature and volume flow rate are also converted to zero.

2) When there are gaps in the raw data of air temperature measurements, these gaps are filled with temperature values of the day before. If that data is not available, they are filled with values from the next day.

3) **Exception:** The sensor *PlateHX_theta_Prim_In (nach Doppelfilter / vor Eingang Schluckbrunnen (Vor-/Rücklauftemperatur))* is a water temperature measurement but the raw data gaps are not filled with zeros but with temperature values of the day before (or day after). This is because it is a measurement that originates from the environment and is used as input data in the script. It is needed continuously.