

Evidence about the linkages set up for GUSTO

Sebastian Zwickl-Bernhard
Energy Economics Group (EEG), TU Wien

Objective

This working document provides an evidence for the linkages set up between the models in **Case Study 3** (namely *openTEPES* and *EMPS-W*) with *GUSTO* (Energy Community System Modeling). In the following, the evidence is provided via (i) read from Scenario Explorer script (ii) write to Scenario Explorer (iii) Results provided from GUSTO to the platform and (iv) the log files after uploading data to the platform. Aggregation and disaggregation tools are included in the underlying Pyam package and therefore, are not mentioned explicitly.

(i) For the results of the model it is necessary to load a) *LoadFactor|Electricity|Solar|Profile* and b) *Price|Final Energy|Residential|Electricity* from the open platform. Since GUSTO considers and models energy systems on a neighborhood level, no further data from the platform (e.g., transmission capacities between European Countries or generation capacity on a national level) are necessary as input for the model. However, the data are used to validate the model results (Final Energy|Residential and Commercial) are from the open platform but not explicitly used as an input.

(ii) The results that are generated from GUSTO and provided to the open platform and other models are different annual time series of a modified electricity profile (*Final Energy|Residential and Commercial|Electricity*).

(iii) Is shown by a result file, including the modified electricity profile for a specific region in Spain (e.g., ES11 - Galicia).

```

# import required packages

import nomenclature
import pyam
from openpyxl import load_workbook

# read data from the scenario explorer
# define data here that should be written to the model input file

_filename = 'Technical_economic_input_data.xlsx'
_worksheet = 'SupIm'
_region = 'Spain|Subgrid 01'
_storyline = 'Societal Commitment'

_df = pyam.read_iiasa('openentrance',
                      model='HEROSCARS v1.0',
                      scenario = 'Baseline',
                      variable = 'LoadFactor|Electricity|Solar|Profile',
                      region = _region
                      )

# _df_elec_prices = pyam.read_iiasa('openentrance',
#                                   model = 'EMPS-W v1.0',
#                                   scenario = _storyline,
#                                   variable =
#                                   'Price|Final Energy|Residential|Electricity',
#                                   region = _region)

_df_elec_prices = _df

# validate the dataframe

if (nomenclature.validate(_df) and nomenclature.validate(_df_elec_prices)):

    print('Dataframe passed *nomenclature.validate*')

    # write input data to .xlsx file
    _wb = load_workbook(_filename)
    # _ws = _wb['_SupIm'] # worksheet including the solar generation profile
    _ws = _wb['_SupIm']

    _counter = 2
    _year = 8760

    while ('.Solar' in str(_ws.cell(row = 1, column = _counter).value)):
        for _i in range(_year):
            _ws.cell(_i+2, _counter).value = _df['value'][_i]
            _counter+=1

    _counter = 2
    _ws = _wb['_Buy-Sell-Price']
    while ('Elec' in str(_ws.cell(row = 1, column = _counter).value)):
        for _i in range(_year):
            _ws.cell(_i+2, _counter).value = _df_elec_prices['value'][_i]
            _counter+=1

    _wb.save(_filename)

```

```

# -*- coding: utf-8 -*-
"""
Created on Tue Oct 13 15:02:39 2020

@author: zwickl-nb
"""

# import required packages
import pandas as pd
import xlrd
import xlswriter
import os
import glob
from datetime import datetime, timedelta

def write_to_iamc_format():

    model_name = 'GUSTO v1.0' # define model name
    _variable = 'Final Energy|Electricity|Profile' # write variable to IAMC
    _unit = 'MW' # define corresponding variable unit

    e = glob.glob(os.path.join('Output', '*')) # get all entries in "Output"
    e.sort(key=lambda x: os.path.getmtime(x)) # sort entries by creation time
    e = e[-1]

    wd = os.path.join(e)
    os.chdir(wd)
    name_files = os.listdir() # get all file names

    # only consider 'scenario*.xlsx' files
    for _n in reversed(range(len(name_files))):
        if not ('scenario_' in name_files[_n] and '.xlsx' in name_files[_n]):
            name_files.pop(_n)

    _scenarios = []
    _regions = []

    for _name in name_files:
        _string = _name.replace(
            'scenario_', '').replace('.xlsx', '').split('+')
        _scenarios.append(_string[0].replace('_', ' '))
        _regions.append(_string[1].replace('_', ' '))

    list_ts=[]

    for _file in name_files:
        worksheet = xlrd.open_workbook(filename=_file, on_demand=True)

        _dict = pd.read_excel(_file, sheet_name=None)
        _to_drop=[]

        for k in _dict.keys():
            if 'timeseries' in k:
                _dict[k] = _dict[k].drop(columns=['Unnamed: 0'])
                _dict[k].columns=_dict[k].loc[0,:].values
                if 'Feed-in public grid' in _dict[k].columns:
                    _dict[k]=_dict[k].iloc[2:,:][
                        'Supply from public grid']-_dict[k].iloc[2:,:][
                            'Feed-in public grid']

```

```

        else:
            _dict[k]=_dict[k].iloc[2:,][['Supply from public grid']]
            pass

    else:
        _to_drop.append(k)

for _d in _to_drop:
    _dict.pop(_d)

list_ts.append(sum(_dict.values()))

# create IAMC-format template
workbook = xlsxwriter.Workbook('GUSTO_results_annual_timeseries.xlsx')
worksheet = workbook.add_worksheet()
bold = workbook.add_format({'bold': True})
worksheet.write('A1', 'model', bold)
worksheet.write('B1', 'scenario', bold)
worksheet.write('C1', 'region', bold)
worksheet.write('D1', 'variable', bold)
worksheet.write('E1', 'unit', bold)
worksheet.write('F1', 'time', bold)
worksheet.write('G1', 'value', bold)

_string = "2030-01-01 00:00"
_var_time = datetime.strptime(_string, '%Y-%m-%d %H:%M')

# write data to IAMC format
for index in range(len(_scenarios)):
    for _entry in range(len(list_ts[0])):
        worksheet.write('A'+str(
            2+_entry+index*len(list_ts[0])), model_name)
        worksheet.write('B'+str(
            2+_entry+index*len(list_ts[0])), _scenarios[index])
        worksheet.write('C'+str(
            2+_entry+index*len(list_ts[0])), _regions[index])
        worksheet.write('D'+str(
            2+_entry+index*len(list_ts[0])), _variable)
        worksheet.write('E'+str(
            2+_entry+index*len(list_ts[0])), _unit)
        worksheet.write('F'+str(
            2+_entry+index*len(list_ts[0])), str(
                _var_time+timedelta(hours=_entry)
            )+' +01:00')
        worksheet.write('G'+str(
            2+_entry+index*len(
                list_ts[0])), list_ts[index].iloc[_entry])

workbook.close()

if __name__ == '__main__':
    write_to_iamc_format()

```

model	scenario	region	variable	unit	time	value
HEROSCARS v1.0	Baseline	ES11	Final Energy Residential and Commercial	MW	2030-01-01 00:00:00 +01:00	235.45
HEROSCARS v1.0	Baseline	ES11	Final Energy Residential and Commercial	MW	2030-01-01 01:00:00 +01:00	-
HEROSCARS v1.0	Baseline	ES11	Final Energy Residential and Commercial	MW	2030-01-01 02:00:00 +01:00	37.34
HEROSCARS v1.0	Baseline	ES11	Final Energy Residential and Commercial	MW	2030-01-01 03:00:00 +01:00	98.82
HEROSCARS v1.0	Baseline	ES11	Final Energy Residential and Commercial	MW	2030-01-01 04:00:00 +01:00	-
HEROSCARS v1.0	Baseline	ES11	Final Energy Residential and Commercial	MW	2030-01-01 05:00:00 +01:00	91.91
HEROSCARS v1.0	Baseline	ES11	Final Energy Residential and Commercial	MW	2030-01-01 06:00:00 +01:00	-
HEROSCARS v1.0	Baseline	ES11	Final Energy Residential and Commercial	MW	2030-01-01 07:00:00 +01:00	79.29
HEROSCARS v1.0	Baseline	ES11	Final Energy Residential and Commercial	MW	2030-01-01 08:00:00 +01:00	-
HEROSCARS v1.0	Baseline	ES11	Final Energy Residential and Commercial	MW	2030-01-01 09:00:00 +01:00	69.35
HEROSCARS v1.0	Baseline	ES11	Final Energy Residential and Commercial	MW	2030-01-01 10:00:00 +01:00	-
HEROSCARS v1.0	Baseline	ES11	Final Energy Residential and Commercial	MW	2030-01-01 11:00:00 +01:00	-
HEROSCARS v1.0	Baseline	ES11	Final Energy Residential and Commercial	MW	2030-01-01 12:00:00 +01:00	64.40
HEROSCARS v1.0	Baseline	ES11	Final Energy Residential and Commercial	MW	2030-01-01 13:00:00 +01:00	7.43
HEROSCARS v1.0	Baseline	ES11	Final Energy Residential and Commercial	MW	2030-01-01 14:00:00 +01:00	-
HEROSCARS v1.0	Baseline	ES11	Final Energy Residential and Commercial	MW	2030-01-01 15:00:00 +01:00	123.86
HEROSCARS v1.0	Baseline	ES11	Final Energy Residential and Commercial	MW	2030-01-01 16:00:00 +01:00	43.88
HEROSCARS v1.0	Baseline	ES11	Final Energy Residential and Commercial	MW	2030-01-01 17:00:00 +01:00	109.22
HEROSCARS v1.0	Baseline	ES11	Final Energy Residential and Commercial	MW	2030-01-01 18:00:00 +01:00	288.49
HEROSCARS v1.0	Baseline	ES11	Final Energy Residential and Commercial	MW	2030-01-01 19:00:00 +01:00	310.90
HEROSCARS v1.0	Baseline	ES11	Final Energy Residential and Commercial	MW	2030-01-01 20:00:00 +01:00	132.25
HEROSCARS v1.0	Baseline	ES11	Final Energy Residential and Commercial	MW	2030-01-01 21:00:00 +01:00	94.67
HEROSCARS v1.0	Baseline	ES11	Final Energy Residential and Commercial	MW	2030-01-01 22:00:00 +01:00	38.08
HEROSCARS v1.0	Baseline	ES11	Final Energy Residential and Commercial	MW	2030-01-01 23:00:00 +01:00	114.43



Job details



ID:	306
Created At:	2020-11-16 05:53
Started At:	2020-11-16 05:53
Finished At:	2020-11-16 05:54
Status:	FINISHED
Created By:	sebastianzwickl

[Details](#)[Log](#)

Download logfile

```
2020-11-16 04:54:15,502 [MainThread ] [INFO ] Downloading 1605502421888-Residential_elec_profile_ES11.xlsx...
2020-11-16 04:54:15,660 [MainThread ] [INFO ] Downloaded file /work/1605502421888-Residential_elec_profile_ES11.xlsx (size: 416603 bytes)!
2020-11-16 04:54:15,660 [MainThread ] [INFO ] Downloaded file /work/1605502421888-Residential_elec_profile_ES11.xlsx (size: 416603 bytes)!
2020-11-16 04:54:15,661 [MainThread ] [INFO ] Start processing for /work/1605502421888-Residential_elec_profile_ES11.xlsx...
2020-11-16 04:54:16,712 [MainThread ] [INFO ] Reading file `work/1605502421888-Residential_elec_profile_ES11.xlsx`
2020-11-16 04:54:20,250 [MainThread ] [WARNI] no meta_scenario data found
2020-11-16 04:54:20,251 [MainThread ] [INFO ] Importing import_timeseries.applications.openentrance.workflow
2020-11-16 04:54:23,302 [MainThread ] [INFO ] Starting openENTRANCE timeseries-upload processing workflow...
2020-11-16 04:54:23,471 [MainThread ] [INFO ] Re-casting from "time" column to "subannual" format
2020-11-16 04:54:27,012 INFO at.ac.iiasa.ixmp.Platform:146 - Welcome to the IX modeling platform!
2020-11-16 04:54:27,017 INFO at.ac.iiasa.ixmp.Platform:147 - connected to database 'jdbc:oracle:thin:@gp3.iiasa.ac.at:1521:GP3' (user: ixmp
2020-11-16 04:54:29,950 [MainThread ] [INFO ] Process scenario "Baseline" model "HEROSCARS v1.0"...
2020-11-16 04:54:29,954 [MainThread ] [INFO ] Create new scenario version...
2020-11-16 04:54:40,964 [MainThread ] [INFO ] Updating meta using the `meta` sheet...
2020-11-16 04:54:40,964 [MainThread ] [INFO ] The `meta` sheet is empty. Continue..
2020-11-16 04:54:40,964 [MainThread ] [INFO ] No `meta_scenario` sheet entries for scenario Baseline found. Continue..
2020-11-16 04:54:40,964 [MainThread ] [INFO ] 1 scenario uploaded to the openentrance database
2020-11-16 04:54:41,339 [MainThread ] [INFO ] Cleaning up caches...
2020-11-16 04:54:41,416 [MainThread ] [INFO ] Job processing finished!
```

[Close](#)