Implementing the DDI-CDI Process Model for Describing Data Integration: Insights from the EOSC Future Science Project “Climate Neutral and Smart Cities”

Benjamin Beuster, Sikt - Norwegian Agency for Shared Services in Education
Joachim Wackerow

EDDI 2023, 15th DDI European User Conference, Ljubljana, November 27-29, 2023

Sikt

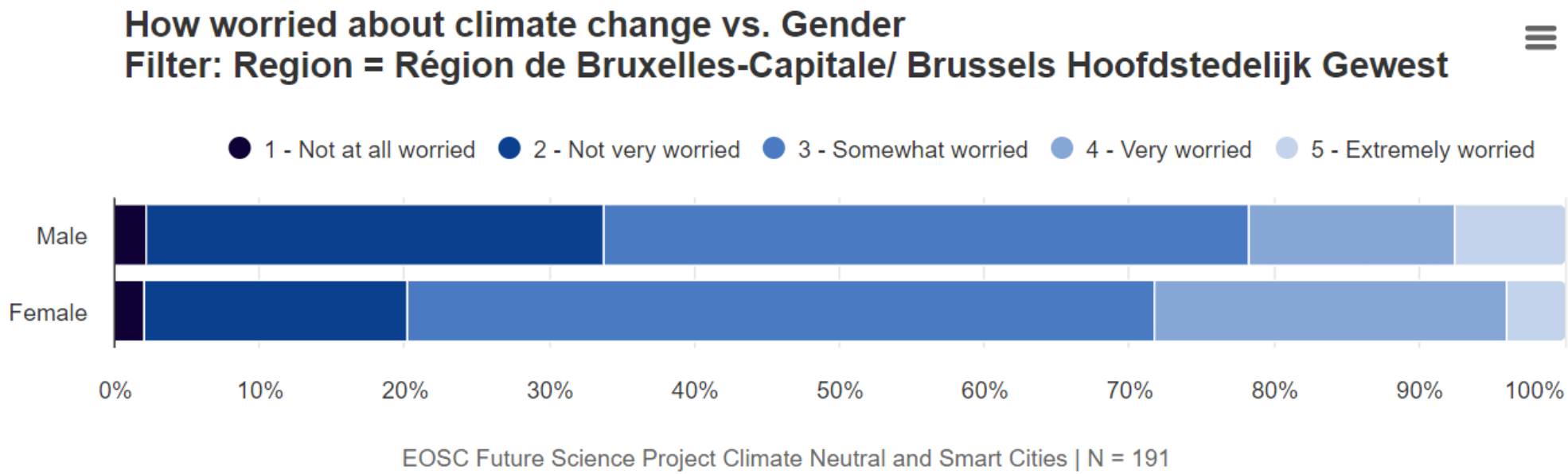
Outline

- EOSC Future Science Project “Climate Neutral and Smart Cities”
- DDI-CDI Process Model - Subset
- Provenance Tool – Live Demonstration

Climate Neutral and Smart Cities project

Goal: Demonstrate that relevant environmental data and data on citizens' values, attitudes, behavior and involvement can be combined for social, political and scientific analysis

ess.sikt.no



ess.sikt.no

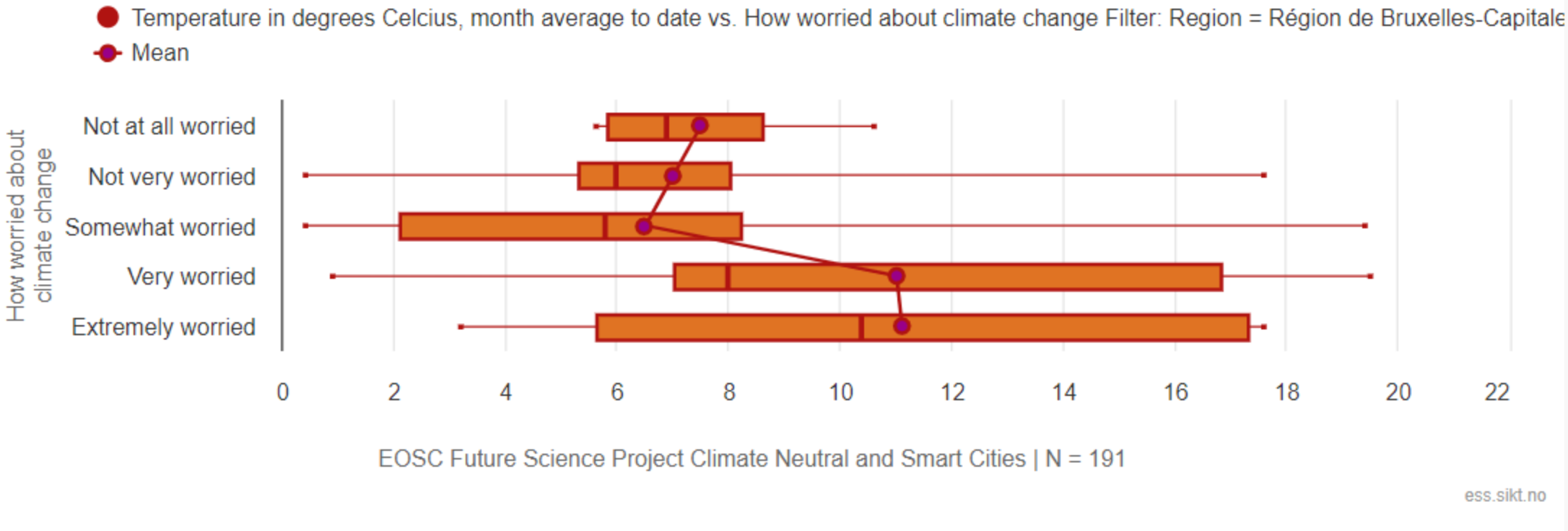
How worried about climate change vs. Gender
Filter: Region = Région de Bruxelles-Capitale/ Brussels Hoofdstedelijk Gewest

	1 - Not at all worried	2 - Not very worried	3 - Somewhat worried	4 - Very worried	5 - Extremely worried	Sum	Mean	N
Male	2.2	31.5	44.6	14.1	7.6	100	2.9	92
Female	2.0	18.2	51.5	24.2	4.0	100	3.1	99


F
C
120
100
80
60
40
20
0
-20
-40
50
40
30
20
10
0
-10
-20
-30
-40
Do you think that the world's climate is changing?

Sikt

Temperature in degrees Celcius, month average to date vs. How worried about climate change
Filter: Region = Région de Bruxelles-Capitale/ Brussels Hoofdstedelijk Gewest



What are the Best Practices for Documenting the Data and the Data Integration Process?

Climate and Air Quality Indices for the ESS - Design spreadsheet for variables

	A	B	C	D	E	F	G
1	**Target Variable Name**	**Target Measure Variable Label**	**Target Measure Variable Description**	**Target Measure Variable Representation**	**Unit of Measure Target Variable**	**Source Variable Name(s)**	**Source Variable Label**
2	date	Date		Representation as in SPSS system format		time	hours since 'offset time (in UTC
3						region_id	Nuts 2016 region code
4	tmpdca	Temperature in degrees Celcius, date average	Regional average daily air temperature at 2m height, for 2016-2022.	Numeric representation, Decimal, min -90, max 90	°C	tmpdc	2 metre temperature
5						region_id	Nuts 2016 region code
6						date	Date
7						pop	Estimated population in grid cell
8	tmpdcmx	Temperature in degrees Celcius, date maximum	Regional average daily maximum air temperature at 2m height, for 2016-2022.	Numeric representation, Decimal, min -90, max 90	°C	tmpdc	2 metre temperature
9						region_id	Nuts 2016 region code
10						date	Date
11						pop	Estimated population in grid cell

Introduction | Air quality indicators | **Climate indicators**

Sikt

Variable Documentation

- Target and Source Data Files
- Target and Source Variables
- Name and Label
- Representation
- Codelists, Missing Values
- Unit of Measure, Description
- Variable Groups

But How Can We Describe the Data Integration Process and Variable Computation?

Process Model

Fully qualified package name: DDICDIModels::DDICDILibrary::Classes::Process

This package contains classes for describing the high-level processes and their substeps ative) and non-linear (non-deterministic, rule-based) processes.

Activity
AllenIntervalAlgebra
ConditionalControlLogic
ControlLogic
Curator
DeterministicImperative
InformationFlowDefinition
NonDeterministicDeclarative
Parameter
ProcessingAgent
ProductionEnvironment
Rule
RuleBasedScheduling
RuleSet
Sequence
Service
Step
TemporalConstraints
TemporalControlConstruct

Process Model

DDICDILibrary

Fully qualified package name: DDICDIModels::DDICDILibrary

This package contains the classes, datatypes, and their definitions for all of the DDI-CDI model packages, as described below.

- Classes
  - Agents
    - Agent
    - Organization
  - Process
    - Activity
    - ControlLogic
    - DeterministicImperative
    - Parameter
    - ProcessingAgent
    - ProductionEnvironment
    - Sequence
    - Step

Subset of Process Model

Sikt

DDICDILibrary

Fully qualified package name: DDICDIModels::DDICDILibrary

This package contains the classes, datatypes, and their definitions for all of the DDI-CDI model packages, as described below.

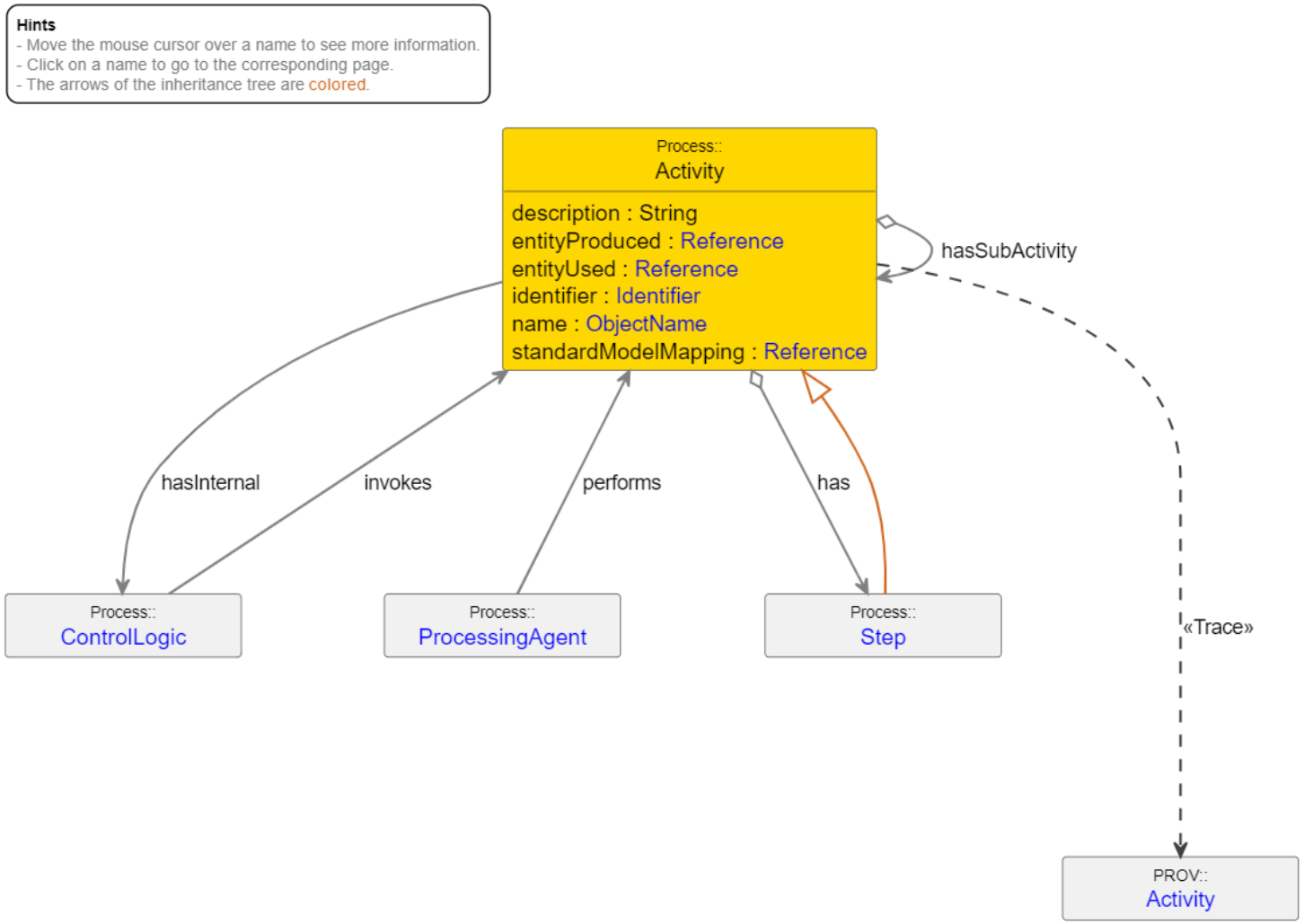
- Classes
  - Agents
    - Agent
    - Organization
  - Process
    - Activity
    - ControlLogic
    - DeterministicImperative
    - Parameter
    - ProcessingAgent
    - ProductionEnvironment
    - Sequence
    - Step

Subset of Process Model

Sikt

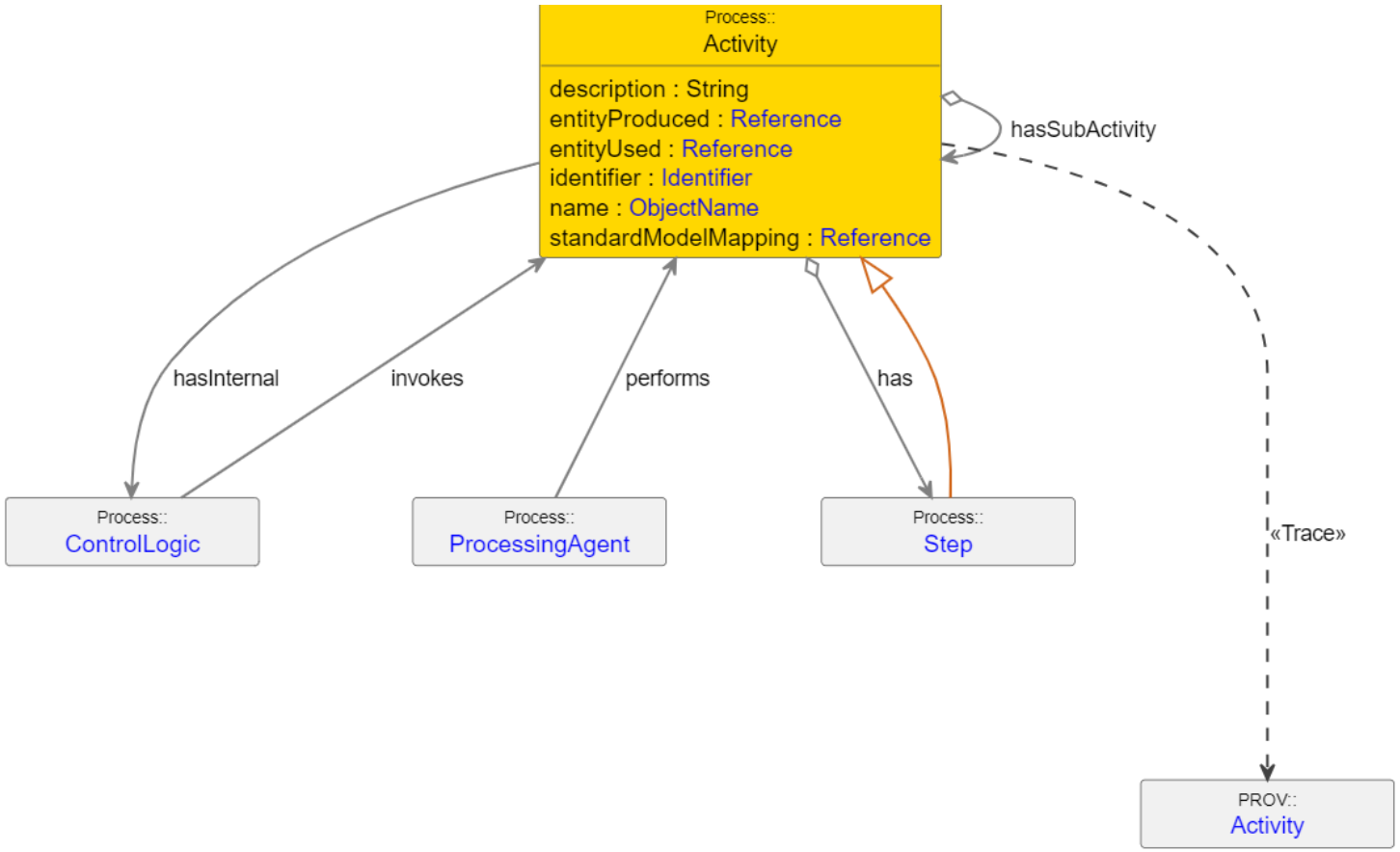
Activity

UML Diagram: Class Activity in Context



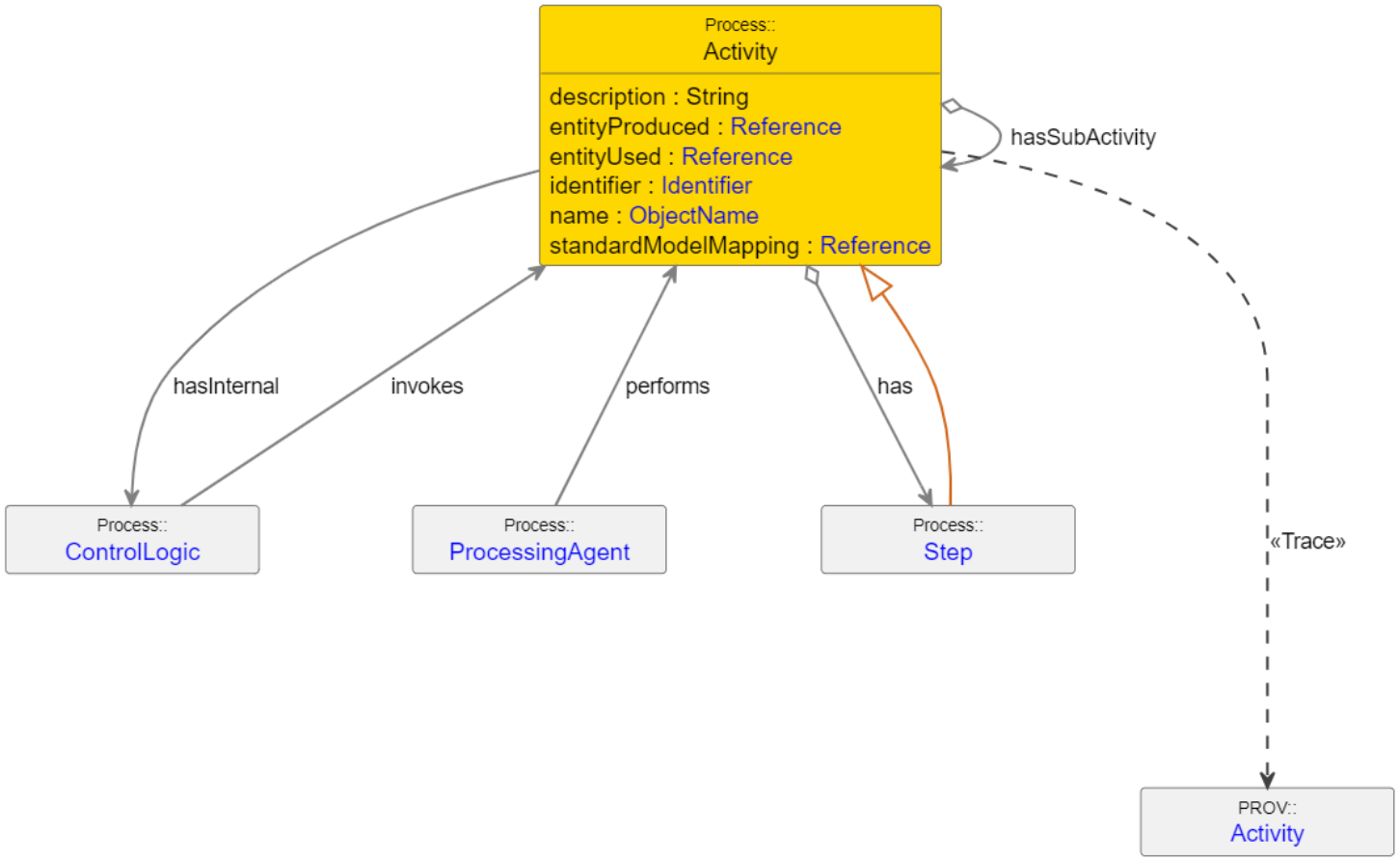
Sikt

Serve as high-level overviews and can include sub-activities to establish a hierarchical structure. Broader scope, such as the data file level, not parameterized.



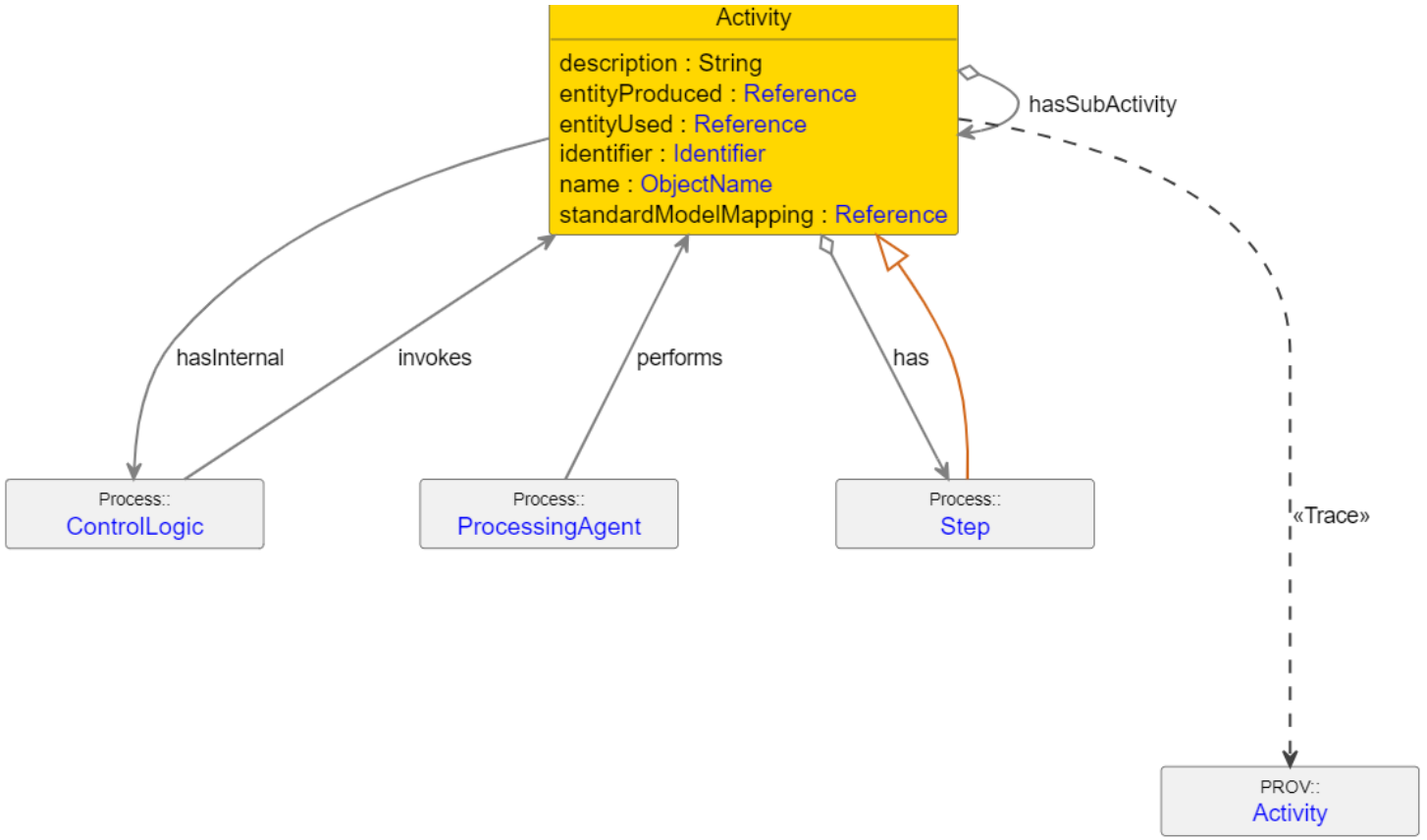
Sikt

Attribute **entityUsed** refers to resources used in the activity,
and **entityProduced** refers to the outcomes resulting from the activity.



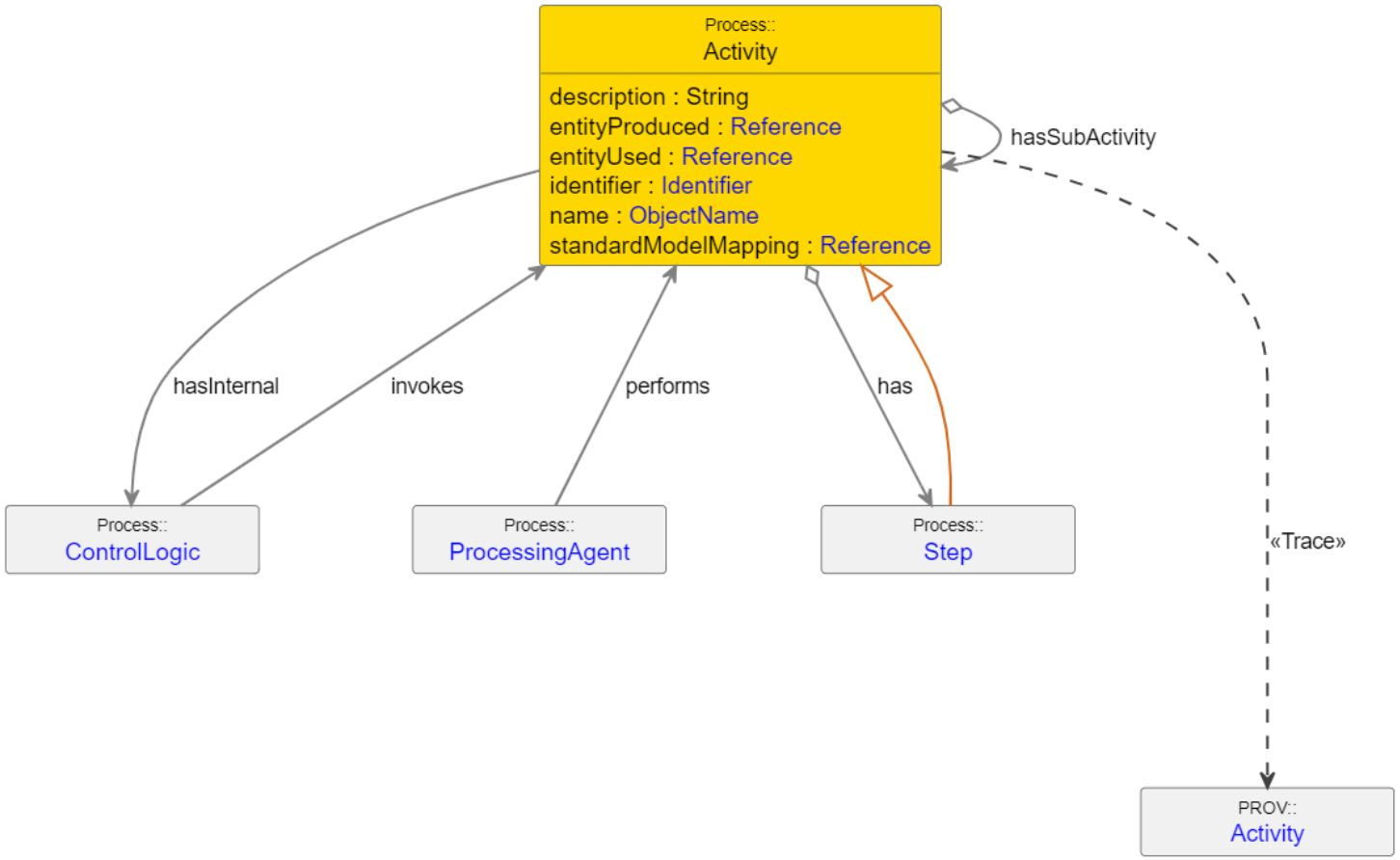
Sikt

Implementation: The resources include different types of web input, such as files available via the internet and APIs, web pages, and source data descriptions.
The outcomes are typically Digital Object Identifiers (DOI) of data files that resolve to landing pages in the ESS Data Portal.

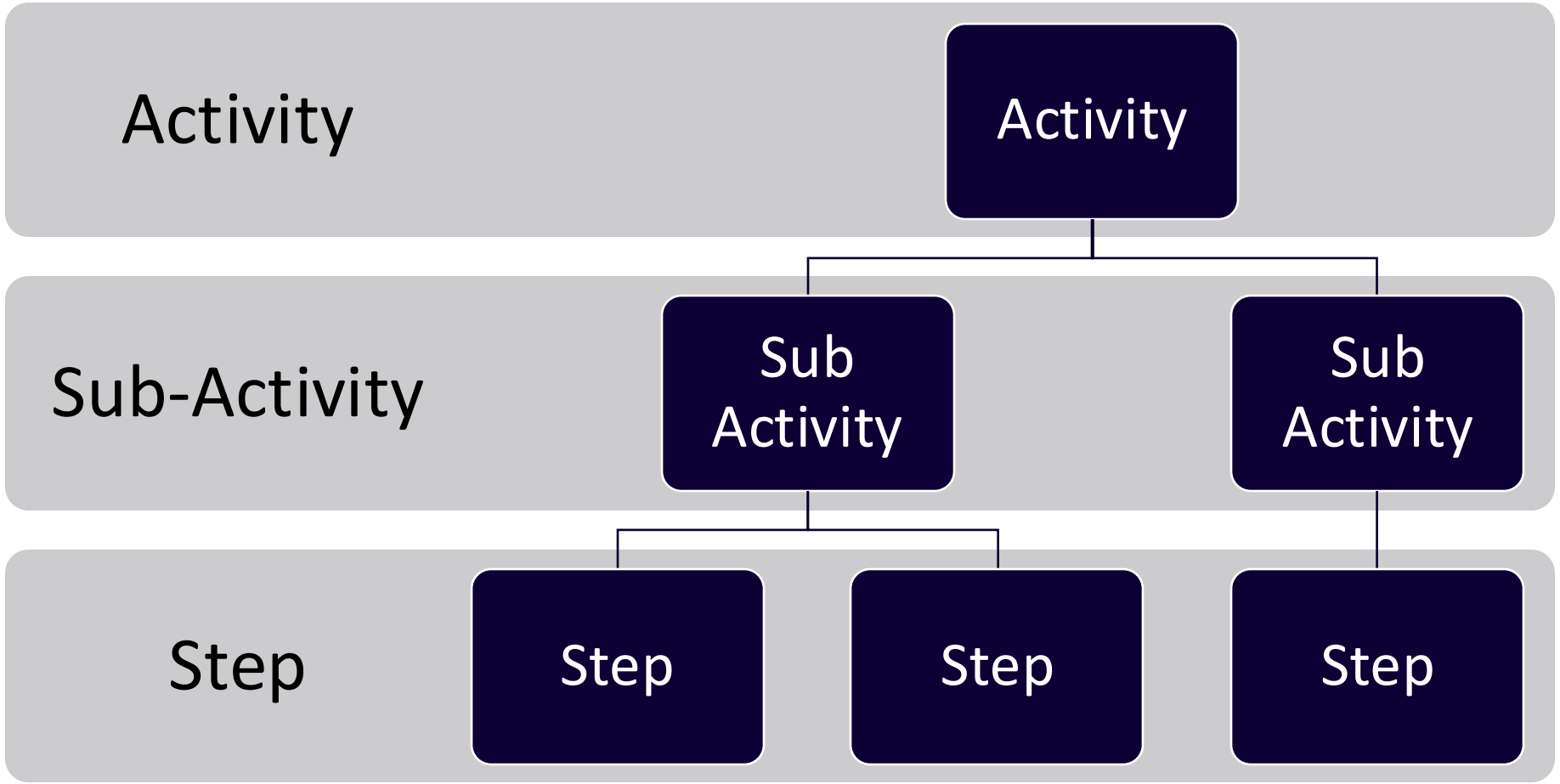


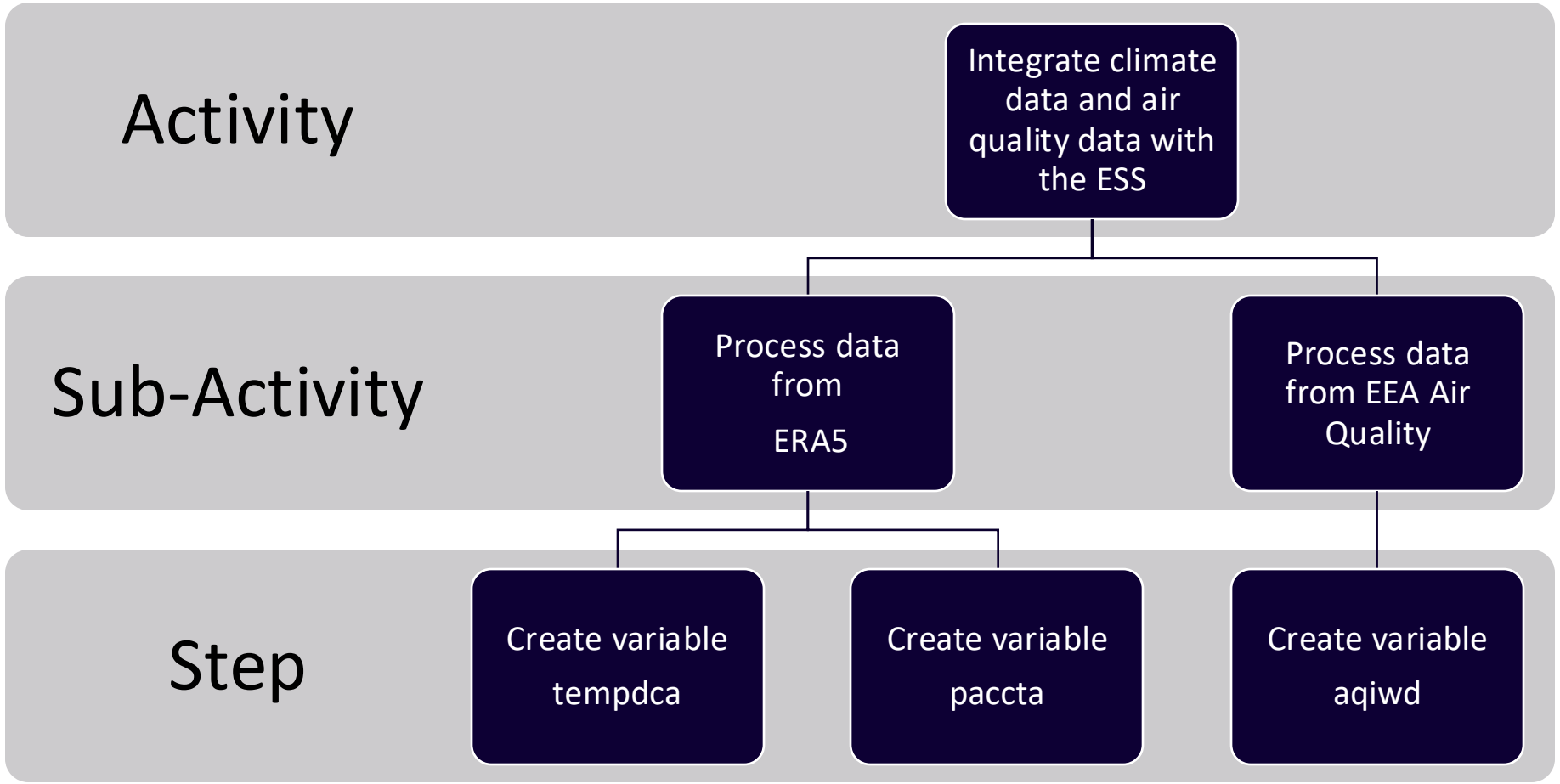
Sikt

Can be broken down into steps for a more detailed account of data processing and variable computation.

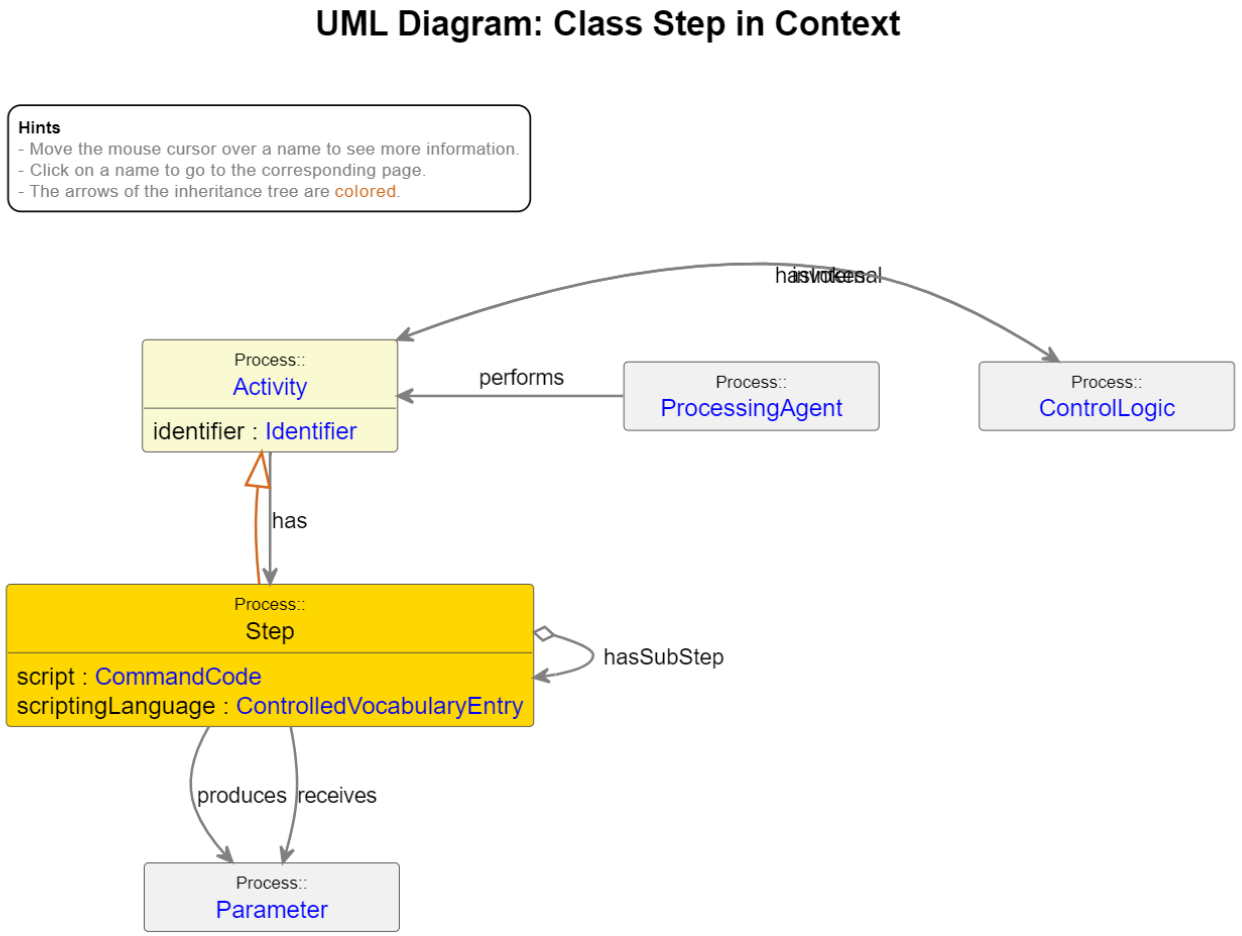


Sikt

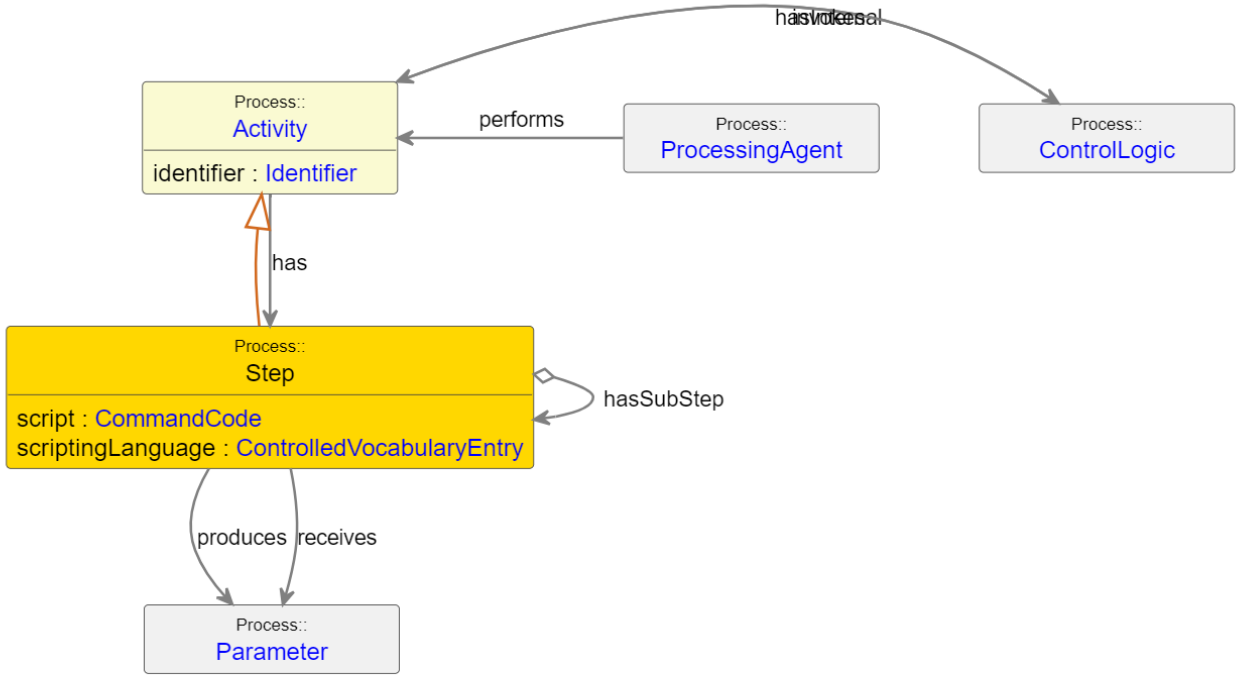

Activity
Sub-Activity
Step
Activity
Sub Activity
Sub Activity
Step
Step
Step


Activity
Integrate climate data and air quality data with the ESS
Sub-Activity
Process data from ERA5
Process data from EEA Air Quality
Step
Create variable tempdca
Create variable paccta
Create variable aqiwd

Step and Parameter

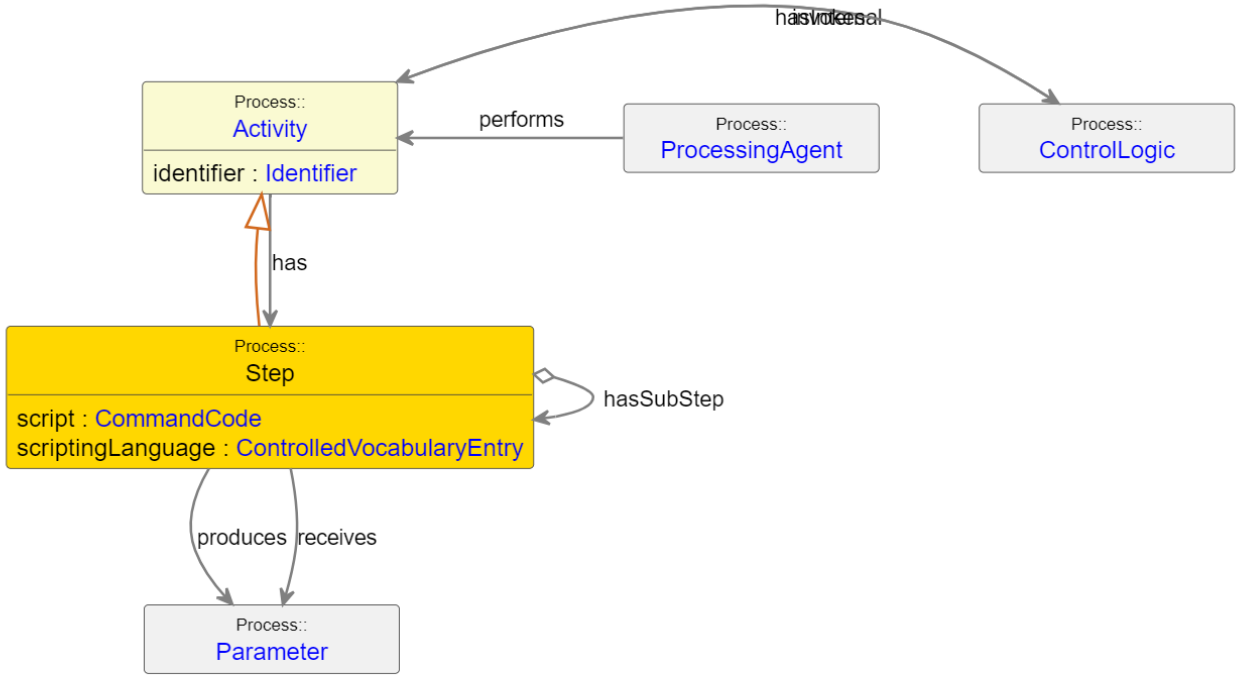


Steps are modular, parameterized sub-processes within an activity that manage the flow of information. They process **input parameters** and generate **output parameters**, which can take various forms like data files or specific variables.



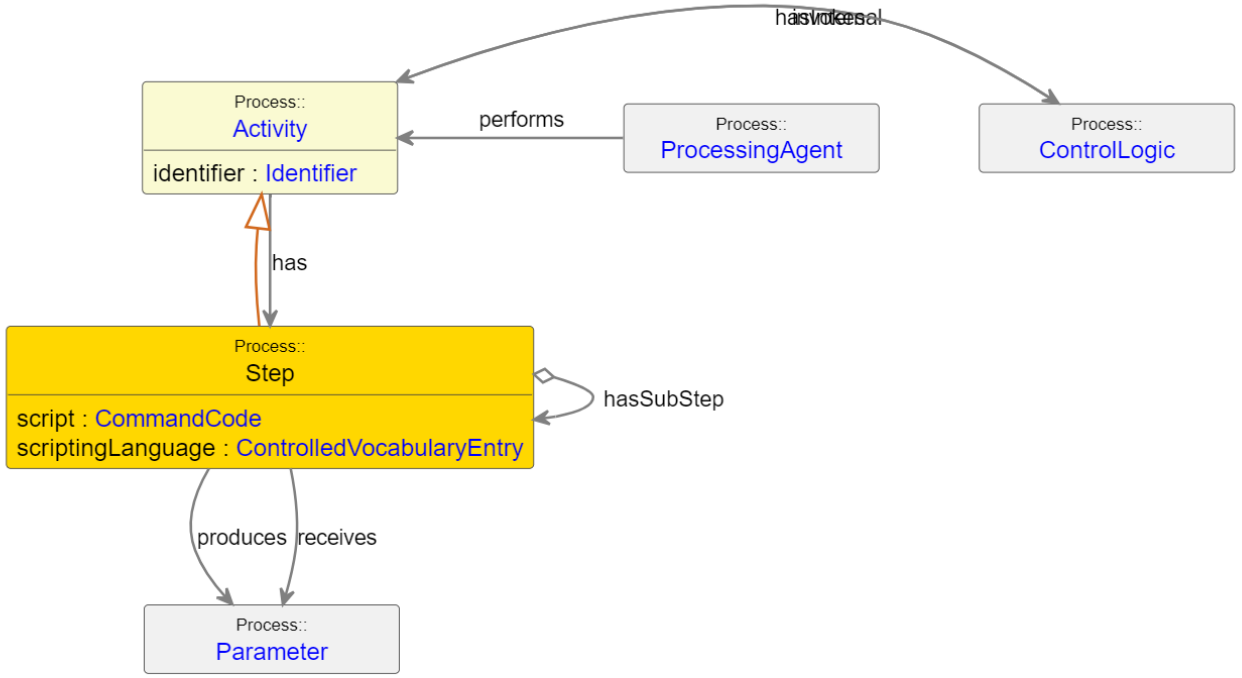
Parameters are tailored to be instance variables, typically represented as columns within a unit record data file.

Implementation: Each parameter has an attribute named **entityBound** which contains a URI directing to the variable's display location in the ESS Data Portal (DDI-L).



The **Script** attribute within the step holds details about the command used for data processing. This can include a description of the command or a URI linking to an external command script.

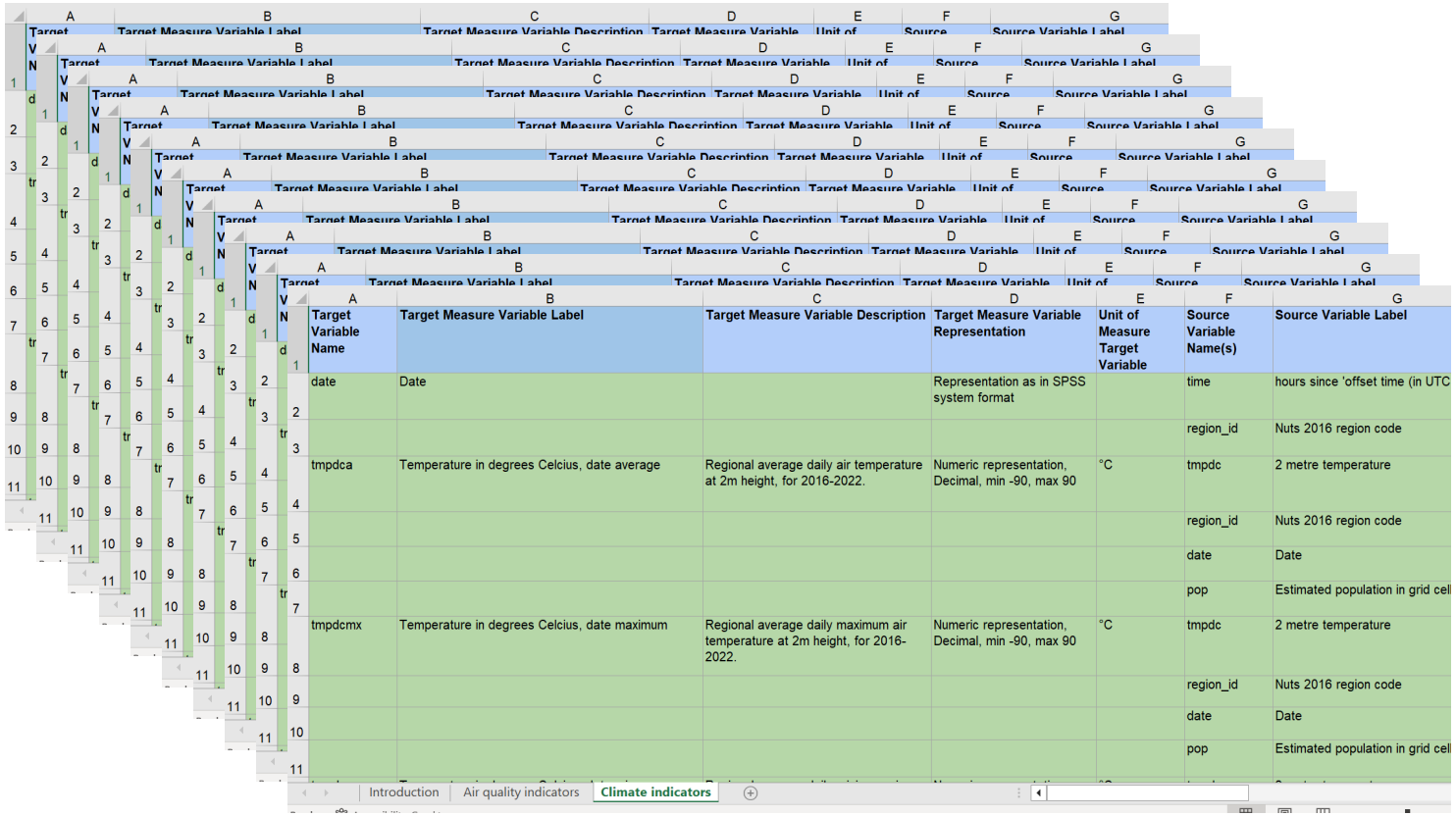
Implementation: the URI leads to a specific Python file in a GitHub repository and pinpoints the exact line where the output parameter is computed.



How to generate the DDI-CDI metadata?

For each class of the subset, we have created a dedicated table

- Classes
  - Agents
    - Agent
    - Organization
  - Process
    - Activity
    - ControlLogic
    - DeterministicImperative
    - Parameter
    - ProcessingAgent
    - ProductionEnvironment
    - Sequence
    - Step

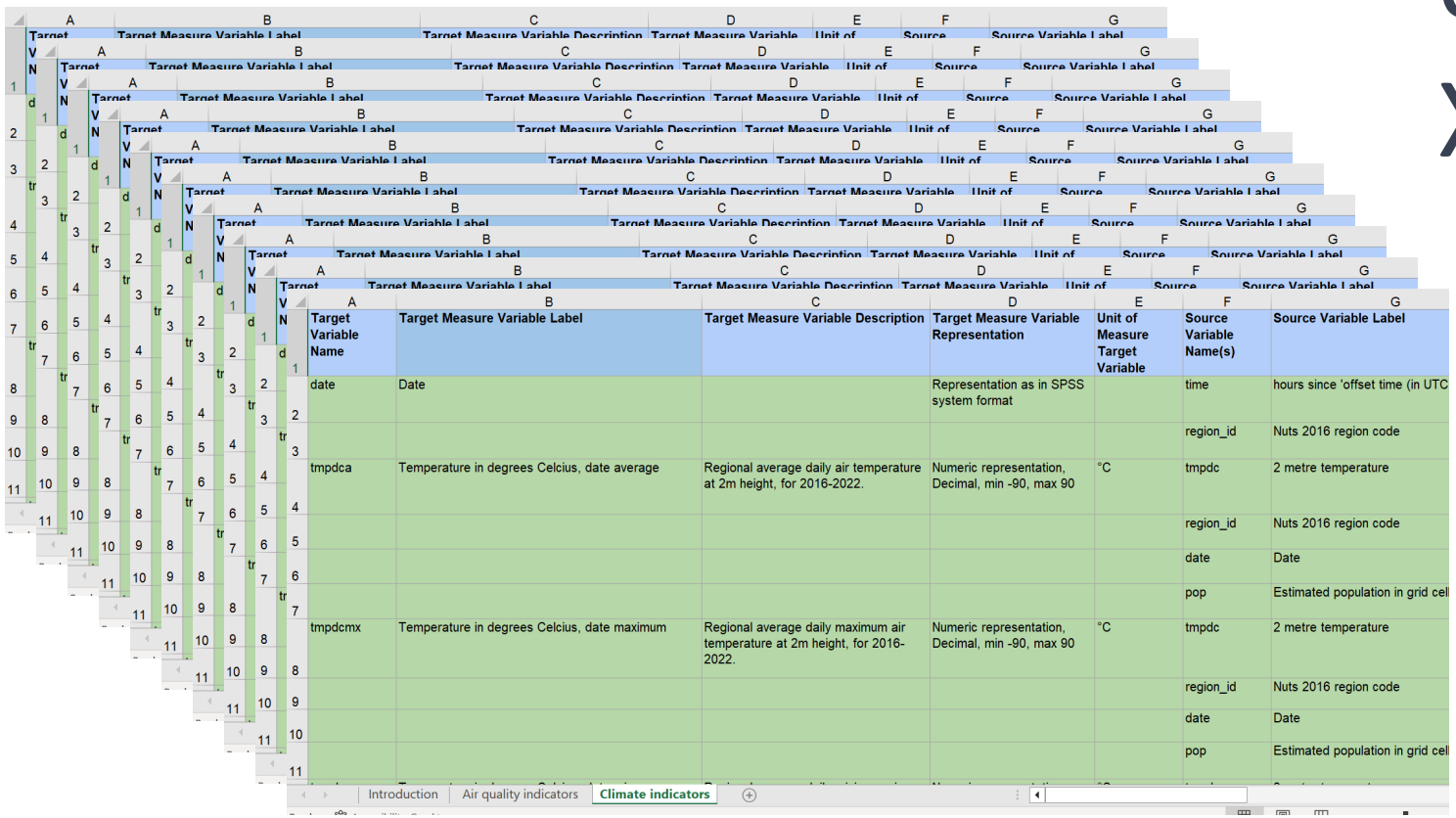


Target Variable Name	Target Measure Variable Label	Target Measure Variable Description	Target Measure Variable Representation	Unit of Measure Target Variable	Source Variable Name(s)	Source Variable Label
date	Date		Representation as in SPSS system format		time	hours since 'offset time (in UTC
					region_id	Nuts 2016 region code
tmpdca	Temperature in degrees Celcius, date average	Regional average daily air temperature at 2m height, for 2016-2022.	Numeric representation, Decimal, min -90, max 90	°C	tmpdc	2 metre temperature
					region_id	Nuts 2016 region code
					date	Date
					pop	Estimated population in grid cell
tmpdcmx	Temperature in degrees Celcius, date maximum	Regional average daily maximum air temperature at 2m height, for 2016-2022.	Numeric representation, Decimal, min -90, max 90	°C	tmpdc	2 metre temperature
					region_id	Nuts 2016 region code
					date	Date
					pop	Estimated population in grid cell

Introduction | Air quality indicators | Climate indicators

Sikt

Each table was then converted into CDI-XML using Python



Target Variable Name	Target Measure Variable Label	Target Measure Variable Description	Target Measure Variable Representation	Unit of Measure Target Variable	Source Variable Name(s)	Source Variable Label
date	Date		Representation as in SPSS system format		time	hours since 'offset time (in UTC
					region_id	Nuts 2016 region code
tmpdca	Temperature in degrees Celcius, date average	Regional average daily air temperature at 2m height, for 2016-2022.	Numeric representation, Decimal, min -90, max 90	°C	tmpdc	2 metre temperature
					region_id	Nuts 2016 region code
					date	Date
					pop	Estimated population in grid cell
tmpdcmx	Temperature in degrees Celcius, date maximum	Regional average daily maximum air temperature at 2m height, for 2016-2022.	Numeric representation, Decimal, min -90, max 90	°C	tmpdc	2 metre temperature
					region_id	Nuts 2016 region code
					date	Date
					pop	Estimated population in grid cell

Introduction | Air quality indicators | Climate indicators

Python

DDI CDI XML

Sikt


Outline
Element name filter
cdi:Activity The process involves
cdi:Activity The process involves
cdi:Activity The process involves
cdi:Activity The process involves
cdi:Step Use variables 'DatetimeE
cdi:Step Compute target variable
cdi:description Compute targe
cdi:identifier
cdi:name Create variable aqiw
cdi:script
cdi:commandFile https://g
cdi:scriptingLanguage Python3
cdi:Step_produces_Parameter-
cdi:ddiReference 83ef6dcf
cdi:dataIdentifier 83ef
cdi:registrationAuthorit
cdi:versionIdentifier 1
CDI-Workflow description EOSC Future.xml
cdi:DDICDIModels
cdi:Step
cdi:name
751 <cdi:isDdiIdentifierUniversallyUnique>true</cdi:isDdiIdentifierUniversallyUnique>
752 </cdi:identifier>
753 <cdi:name>
754 <cdi:name>Create variable aqiwdpm10</cdi:name>
755 </cdi:name>
756 <cdi:script>
757 <cdi:commandFile>
758 <cdi:uri>https://github.com/sikt-no/ess-labs-data-sp9/blob/master/eea-prepare.py#L71</cdi:uri>
759 </cdi:commandFile>
760 </cdi:script>
761 <cdi:scriptingLanguage>
762 <cdi:entryValue>Python3</cdi:entryValue>
763 </cdi:scriptingLanguage>
764 <cdi:Step_produces_Parameter-Target>
765 <cdi:ddiReference><cdi:dataIdentifier>83ef6dcf-2d56-4fd6-94e1-8cbbdca65f55</cdi:dataIdentifier><cdi:registr
766 </cdi:Step_produces_Parameter-Target>
767 <cdi:Step_receives_Parameter-Target>
768 <cdi:ddiReference><cdi:dataIdentifier>1b87f6c7-d1ce-4321-8e76-f94a16e780b6</cdi:dataIdentifier><cdi:registr
769 </cdi:Step_receives_Parameter-Target>
770 <cdi:Step_receives_Parameter-Target>
771 <cdi:ddiReference><cdi:dataIdentifier>2eddc4ae-156e-4b85-8449-09487a26c0a0</cdi:dataIdentifier><cdi:registr
772 </cdi:Step_receives_Parameter-Target>
773 <cdi:Step_receives_Parameter-Target>
774 <cdi:ddiReference><cdi:dataIdentifier>1c576b4d-0eaa-4542-8c17-4b74ff1a854b</cdi:dataIdentifier><cdi:registr
775 </cdi:Step_receives_Parameter-Target>

Sikt

Tour of the tool

https://eosc-provenance.sikt.no/#

Accreditation

The following sections are authored by the Hilde Orten, based on materials developed under the EOSC Science Project 9.

- Slide 2, 3, 5

The content has been adapted for the purpose of this presentation

Thank you!

https://eosc-provenance.sikt.no/#
https://ess.sikt.no
benjamin.beuster@sikt.no
joachim.wackerow@posteo.de

European Social Survey

ESS Labs Process Search

Go

Contents

CDI-Workflow description of the EOSC Future WP6 Task 3, Science Project 9 'Climate Neutral and Smart Cities'

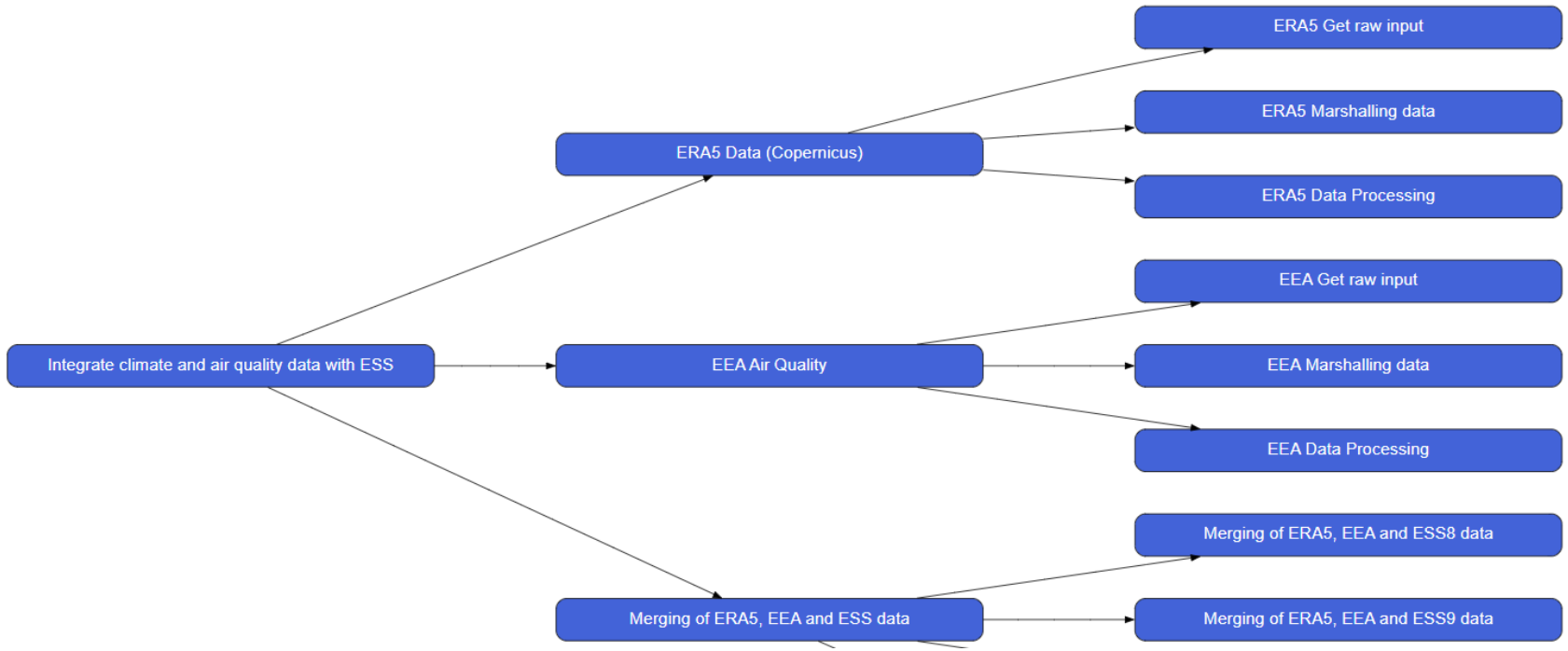
Main Process Sequence

Description: Main Sequence of the process

Processing Agent: EOSC project team at Sikt - Norwegian Agency for Shared Services in Education and Research
Purpose: Integrate climate data from ERA5 and air quality data from the EEA with the ESS survey data
Production Environment: Sikt - Norwegian Agency for Shared Services in Education and Research acting as a participant of SP9

Overview Diagram of the Process Activities (in sequential order)

Note: *Move the mouse cursor over an activity to see more information. Click on an activity to go to the corresponding page.*



Integrate climate and air quality data with ESS

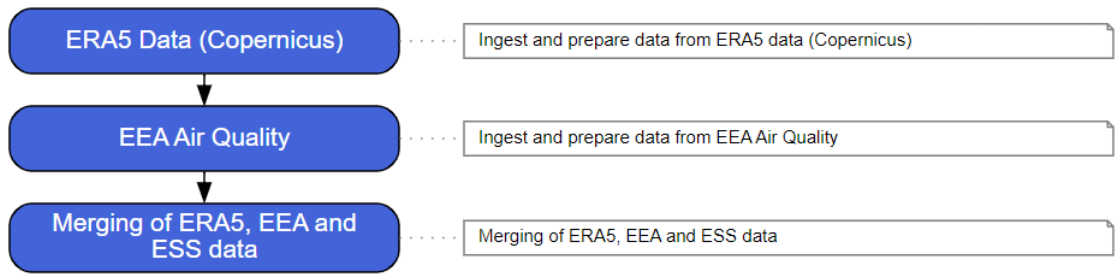
Process Activity

Description: Integrate climate data from Copernicus ERA5 and air quality data from the European Environmental Agency (EEA) with data from the European Social Survey (ESS) for Berlin, Oslo, Stockholm, Brussels, London, Paris, Vienna, Prague, Budapest, and Madrid

---

Diagram of the Process Sub-Activities (in sequential order)

Note: *Click on a sub-activity to go to the corresponding page.*



ERA5 Data (Copernicus)¶

Process Activity

Description: Ingest and prepare data from ERA5 data (Copernicus)

Diagram of the Process Activity

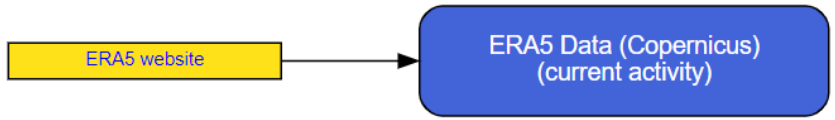
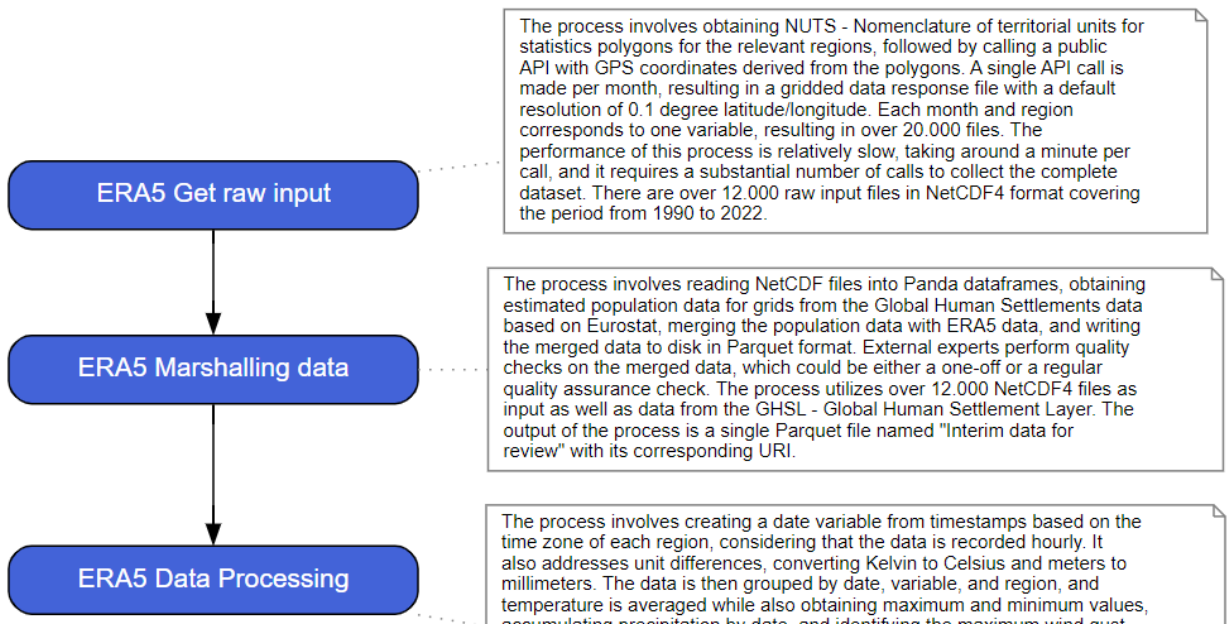


Diagram of the Process Sub-Activities (in sequential order)

Note: *Click on a sub-activity to go to the corresponding page.*

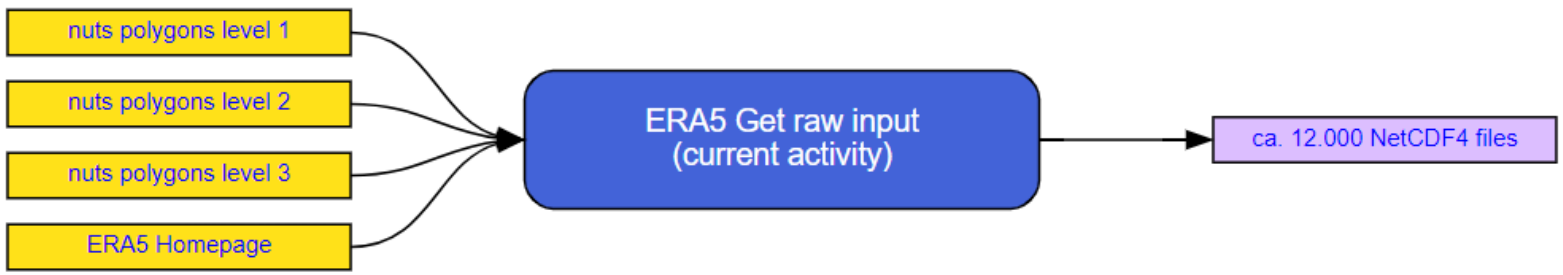


ERA5 Get raw input ¶

Process Activity

Description: The process involves obtaining NUTS - Nomenclature of territorial units for statistics polygons for the relevant regions, followed by calling a public API with GPS coordinates derived from the polygons. A single API call is made per month, resulting in a gridded data response file with a default resolution of 0.1 degree latitude/longitude. Each month and region corresponds to one variable, resulting in over 20.000 files. The performance of this process is relatively slow, taking around a minute per call, and it requires a substantial number of calls to collect the complete dataset. There are over 12.000 raw input files in NetCDF4 format covering the period from 1990 to 2022.

Diagram of the Process Activity



https://gisco-services.ec.europa.eu/distribution/v2/nuts/geojson/NUTS_RG_01M_2016_4326_LEVL_1.geojson

{"type": "FeatureCollection", "features": [{"type": "Feature", "geometry": {"type": "MultiPolygon", "coordinates": [[[[5.682001, 50.757446], [5.689727, 50.75158], [5.704657, 50.750169], [5.721842, 50.74559], [5.727227, 50.751914], [5.73704, 50.755088], [5.745753, 50.752244], [5.745848, 50.747651], [5.748664, 50.745744], [5.771138, 50.750627], [5.77704, 50.7485], [5.787457, 50.744744], [5.794734, 50.738456], [5.801903, 50.73562], [5.801497, 50.73057], [5.810739, 50.72297], [5.814005, 50.715719], [5.819052, 50.714557], [5.820802, 50.713611], [5.8357, 50.71445], [5.842157, 50.716152], [5.853097, 50.716518], [5.858239, 50.717727], [5.865648, 50.716747], [5.868925, 50.714969], [5.874247, 50.713791], [5.882454, 50.710148], [5.910399, 50.735241], [5.905745, 50.742303], [5.894784, 50.747066], [5.891009, 50.751696], [5.892073, 50.755237], [5.901262, 50.752076], [5.905964, 50.755288], [5.913641, 50.755315], [5.917619, 50.750964], [5.930424, 50.756366], [5.936111, 50.756802], [5.948221, 50.759435], [5.960132, 50.762024], [5.971998, 50.759749], [5.975391, 50.755419], [5.98356, 50.75309], [6.020999, 50.754295], [6.039766, 50.745265], [6.04033, 50.738897], [6.033917, 50.729268], [6.037461, 50.719646], [6.041407, 50.719409], [6.044989, 50.727841], [6.059506, 50.724033], [6.07402, 50.721077], [6.090623, 50.722259], [6.102935, 50.722797], [6.116306, 50.720755], [6.123238, 50.712441], [6.125281, 50.705531], [6.142148, 50.68946], [6.145005, 50.681856], [6.160723, 50.671552], [6.166451, 50.662105], [6.183225, 50.650291], [6.194339, 50.657504], [6.19324, 50.664077], [6.196459, 50.659579], [6.195773, 50.656425], [6.178624, 50.644611], [6.167575, 50.643803], [6.18274, 50.632215], [6.180368, 50.628928], [6.176163, 50.627217], [6.176402, 50.625997], [6.182441, 50.624293], [6.188126, 50.627408], [6.187784, 50.629797], [6.194143, 50.633518], [6.20013, 50.631789], [6.217715, 50.631884], [6.235453, 50.625924], [6.251809, 50.62695], [6.261848, 50.628185], [6.268842, 50.625349], [6.268658, 50.619106], [6.261716, 50.611614], [6.248751, 50.604065], [6.248885, 50.597719], [6.241211, 50.588138], [6.225282, 50.589886], [6.219832, 50.582273], [6.207495, 50.575813], [6.202297, 50.569312], [6.189313, 50.566094], [6.175421, 50.557076], [6.178239, 50.553279], [6.178562, 50.542222], [6.18734, 50.540591], [6.197841, 50.53504], [6.196467, 50.531804], [6.188097, 50.527231], [6.192201, 50.521055], [6.206272, 50.520252], [6.221329, 50.503187], [6.221247, 50.497913], [6.225219, 50.495481], [6.251327, 50.503034], [6.259552, 50.499096], [6.269526, 50.503926], [6.280388, 50.502848], [6.285006, 50.498965], [6.297613, 50.497668], [6.30567, 50.500305], [6.309076, 50.501412], [6.314121, 50.499027], [6.314201, 50.497813], [6.315556, 50.497042], [6.330989, 50.492501], [6.335357, 50.489499], [6.348515, 50.488477], [6.341801, 50.483206], [6.338084, 50.474854], [6.343193, 50.468998], [6.341371, 50.461974], [6.350661, 50.457031], [6.36317, 50.454276], [6.371348, 50.45506], [6.374572, 50.451448], [6.377351, 50.438286], [6.375418, 50.43079], [6.367259, 50.419412], [6.369301, 50.40871], [6.35615, 50.391016], [6.343116, 50.380378], [6.360915, 50.37083], [6.369136, 50.358463], [6.397459, 50.345882], [6.400765, 50.33888], [6.407419, 50.334817], [6.405029, 50.323308], [6.383243, 50.321557], [6.373937, 50.313625], [6.361361, 50.314162], [6.36102, 50.312645], [6.359994, 50.308084], [6.355863, 50.309386], [6.350577, 50.312645], [6.342157, 50.317834], [6.335522, 50.318079], [6.333173, 50.322444], [6.327727, 50.323918], [6.318968, 50.319799], [6.31266, 50.321479], [6.307055, 50.31995], [6.308527, 50.312645], [6.308786, 50.311361], [6.296769, 50.308777], [6.29768, 50.302702], [6.286814, 50.293806], [6.286028, 50.284228], [6.29209, 50.278829], [6.278268, 50.266731], [6.276985, 50.266134], [6.271491, 50.267242], [6.26422, 50.264991], [6.257701, 50.267066], [6.246439, 50.262372], [6.235721, 50.261989], [6.227226, 50.258692], [6.223981, 50.258661], [6.22283, 50.257248], [6.215755, 50.256339], [6.206986, 50.25216], [6.200103, 50.242218], [6.197248, 50.238093], [6.178787, 50.236184], [6.173928, 50.231627], [6.176373, 50.226595], [6.16817, 50.2238], [6.166887, 50.220866], [6.177133, 50.216562], [6.187617, 50.205333], [6.184698, 50.192088], [6.189608, 50.18739], [6.187119, 50.183944], [6.192752, 50.181874], [6.184346, 50.178584], [6.164673, 50.178634], [6.161192, 50.174624], [6.157326, 50.172014], [6.1518, 50.176308], [6.14667, 50.17688], [6.146517, 50.170732], [6.140763, 50.168058], [6.146594, 50.161945], [6.146373, 50.15916], [6.142424, 50.156135], [6.13292, 50.154636], [6.140755, 50.148755], [6.148383, 50.151351], [6.152872, 50.15009], [6.153587, 50.141307], [6.148595, 50.136594], [6.139574, 50.134697], [6.138363, 50.131322], [6.137663, 50.129951], [6.133344, 50.129235], [6.121038, 50.135616], [6.112318, 50.13612], [6.115727, 50.144966], [6.123462, 50.150383], [6.115884, 50.156183], [6.119079, 50.163103], [6.104144, 50.169842], [6.090623, 50.170502], [6.080907, 50.170977], [6.078729, 50.157416], [6.064428, 50.154], [6.045673, 50.157884], [6.030557, 50.164961], [6.0249, 50.182779], [6.002163, 50.176242], [5.973001, 50.174987], [5.965035, 50.172066], [5.962644, 50.161557], [5.966567, 50.155308], [5.959303, 50.151336], [5.960671, 50.132187], [5.954156, 50.130198], [5.934154, 50.126845], [5.93071, 50.126267], [5.913777, 50.121316], [5.895269, 50.112216], [5.894632, 50.10191], [5.886613, 50.090589], [5.88673, 50.079294], [5.868096, 50.073142], [5.859939, 50.0681], [5.856644, 50.061633], [5.85742, 50.056182], [5.865328, 50.051783], [5.869229, 50.046849], [5.857275, 50.036172], [5.859123, 50.029301], [5.852732, 50.026874], [5.850139, 50.021957], [5.832963, 50.017856], [5.819579, 50.012213], [5.822796, 49.998512], [5.832431, 49.991077], [5.840233, 49.989372], [5.838439, 49.982418], [5.831974, 49.976985], [5.811387, 49.971147], [5.810069, 49.963869], [5.801483, 49.963203], [5.795005, 49.965172], [5.789012, 49.961526], [5.77704, 49.960947], [5.771535, 49.949129], [5.774102, 49.938697], [5.763863, 49.928975], [5.76335, 49.915419], [5.748186, 49.909472], [5.740653, 49.903957], [5.736842, 49.89684], [5.77704, 49.884055], [5.782788, 49.875775], [5.779795, 49.875498], [5.782431, 49.872464], [5.77704, 49.869386], [5.77446, 49.867914], [5.770134, 49.871643], [5.764283, 49.869986], [5.75518, 49.872034], [5.753573, 49.868749], [5.758969, 49.856547], [5.746319, 49.853595], [5.755537, 49.850969], [5.757976, 49.848243], [5.756348, 49.846059], [5.749659, 49.846052], [5.748631, 49.83954], [5.741812, 49.838557], [5.739997, 49.835658], [5.747955, 49.825129], [5.744255, 49.820177], [5.750407, 49.814108], [5.755378, 49.791433], [5.767902, 49.793909], [5.77704, 49.793191], [5.781138, 49.792868], [5.788057, 49.796119], [5.793448, 49.790546], [5.792918, 49.786546], [5.805793, 49.773996], [5.809631, 49.76608], [5.815384, 49.762217], [5.818208, 49.754568], [5.822754, 49.750174], [5.829573, 49.749127], [5.831383, 49.74525], [5.83114, 49.741191], [5.827482, 49.738378], [5.826179,

Sikt

ERA5 Marshalling data

Process Activity

Description: The process involves reading NetCDF files into Panda dataframes, obtaining estimated population data for grids from the Global Human Settlements data based on Eurostat, merging the population data with ERA5 data, and writing the merged data to disk in Parquet format. External experts perform quality checks on the merged data, which could be either a one-off or a regular quality assurance check. The process utilizes over 12.000 NetCDF4 files as input as well as data from the GHSL - Global Human Settlement Layer. The output of the process is a single Parquet file named “Interim data for review” with its corresponding URI.

Diagram of the Process Activity

European Social Survey

ESS Labs Process Search

Go

Contents

ERA5 Data Processing

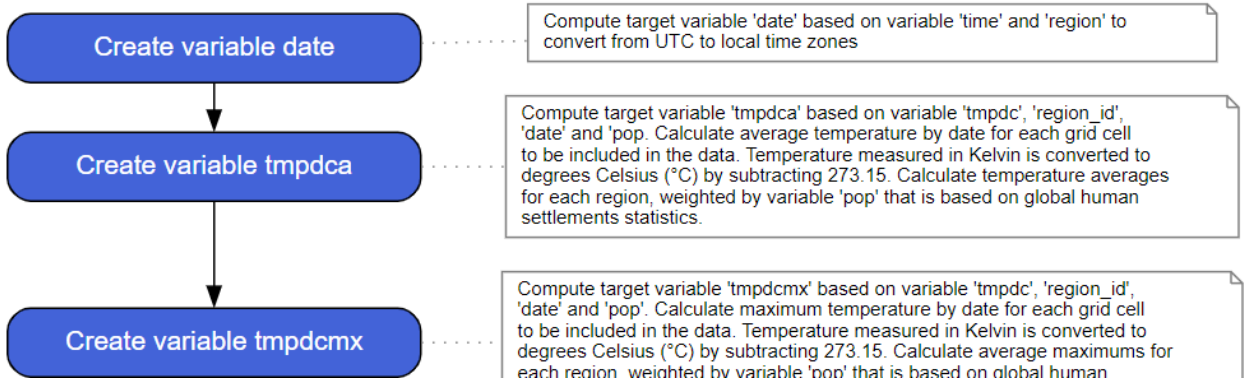
Process Activity

Description: The process involves creating a date variable from timestamps based on the time zone of each region, considering that the data is recorded hourly. It also addresses unit differences, converting Kelvin to Celsius and meters to millimeters. The data is then grouped by date, variable, and region, and temperature is averaged while also obtaining maximum and minimum values, accumulating precipitation by date, and identifying the maximum wind gust value. Moving averages are calculated for variables using different time windows (7-day, 30-day, 90-day, 365-day). Baseline values for temperature, precipitation, wind gust, and deviations from the baseline (anomalies) are determined based on the period from 1991 to 2020. Data older than 2015 is removed, and a group-by operation is performed, collapsing the data by region using population-weighted averages. It is important to note that the ERA5 data may contain imputed and missing values. In memory, each row corresponds to a region, with mesh-blocks aggregated per day to calculate region-level values by taking the average of all variables weighted by the population of each block. The resulting data is stored to disk in CSV, SAV, or other suitable formats, as the data size remains manageable.

Diagram of the Process Activity

Diagram of the Process Sub-Activities (in sequential order)

Note: *Click on a sub-activity to go to the corresponding page.*



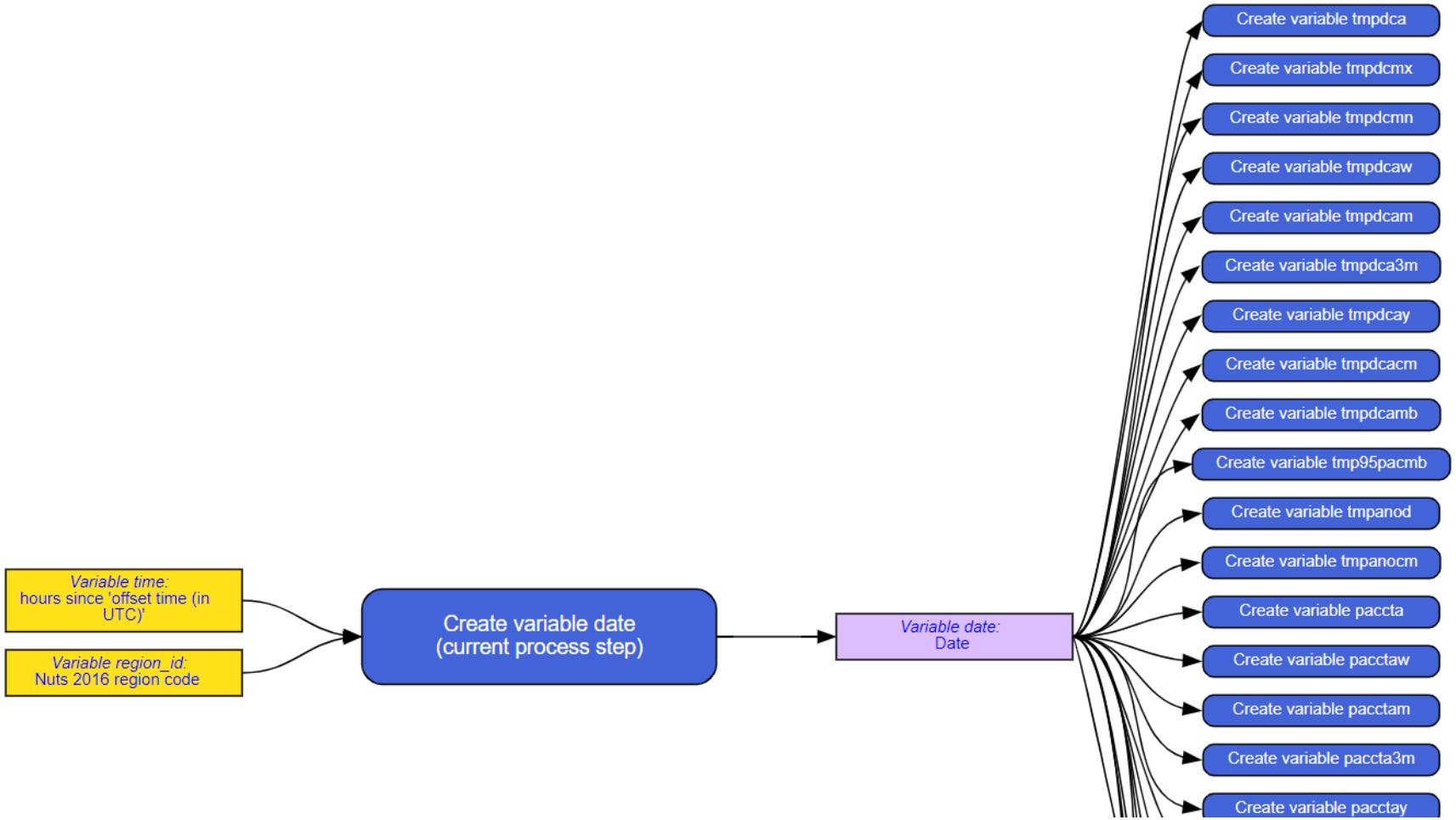
Create variable date ¶

Process Step

Description Compute target variable ‘date’ based on variable ‘time’ and ‘region’ to convert from UTC to local time zones

This step uses a script written in Python3.

Diagram of the Process Step



Sikt

Create variable tmpdca

Process Step

Description Compute target variable ‘tmpdca’ based on variable ‘tmpdc’, ‘region_id’, ‘date’ and ‘pop. Calculate average temperature by date for each grid cell to be included in the data. Temperature measured in Kelvin is converted to degrees Celsius (°C) by subtracting 273.15. Calculate temperature averages for each region, weighted by variable ‘pop’ that is based on global human settlements statistics.

This step uses a script written in Python3.

Diagram of the Process Step

Hint: *Move the mouse cursor over a parameter to see more information. Click on a parameter or a related step to go to the corresponding page.*

Legend: Input parameter — used by → Process step — produces → Output parameter

https://github.com/sikt-no/ess-labs-data-sp9/blob/master/era5-prepare.py#L41

Files

master

Go to file

- .gitignore
- README.md
- config_RENAME_ME.py
- eea-download.py
- eea-prepare.py
- era5-download.py
- era5-prepare.py
- merge.py
- requirements.in
- requirements.txt
- utils.py

ess-labs-data-sp9 / era5-prepare.py

Code | Blame | 265 lines (214 loc) · 8.16 KB

```
14      def create_date_column(df):
28          df["tmpdc"] = df["tmpdc"] - 273.15  # kelvin to celcius
29          df["pac"] = (df["pac"] * 1000).round(2)  # meters to millimeters
30          return df
31
32
33      def groupby_date(df_in: pd.DataFrame) -> pd.DataFrame:
34          """
35          Calculate grid-based daily values
36          """
37          daily_grouper = df_in.groupby(["region", "grid_id", "date"])
38          df = pd.DataFrame(
39              {
40                  "pop": daily_grouper["pop"].first(),
41                  "tmpdca": daily_grouper["tmpdc"].mean(numeric_only=True),
42                  "tmpdcmx": daily_grouper["tmpdc"].max(),
43                  "tmpdcmn": daily_grouper["tmpdc"].min(),
44                  "paccta": daily_grouper["pac"].sum(),
45                  "iwg10mx": daily_grouper["iwg10"].max(),
46              }
47          )
48          return df
49
```

Sikt

Create variable tmpdca

Process Step

Description Compute target variable ‘tmpdca’ based on variable ‘tmpdc’, ‘region_id’, ‘date’ and ‘pop. Calculate average temperature by date for each grid cell to be included in the data. Temperature measured in Kelvin is converted to degrees Celsius (°C) by subtracting 273.15. Calculate temperature averages for each region, weighted by variable ‘pop’ that is based on global human settlements statistics.

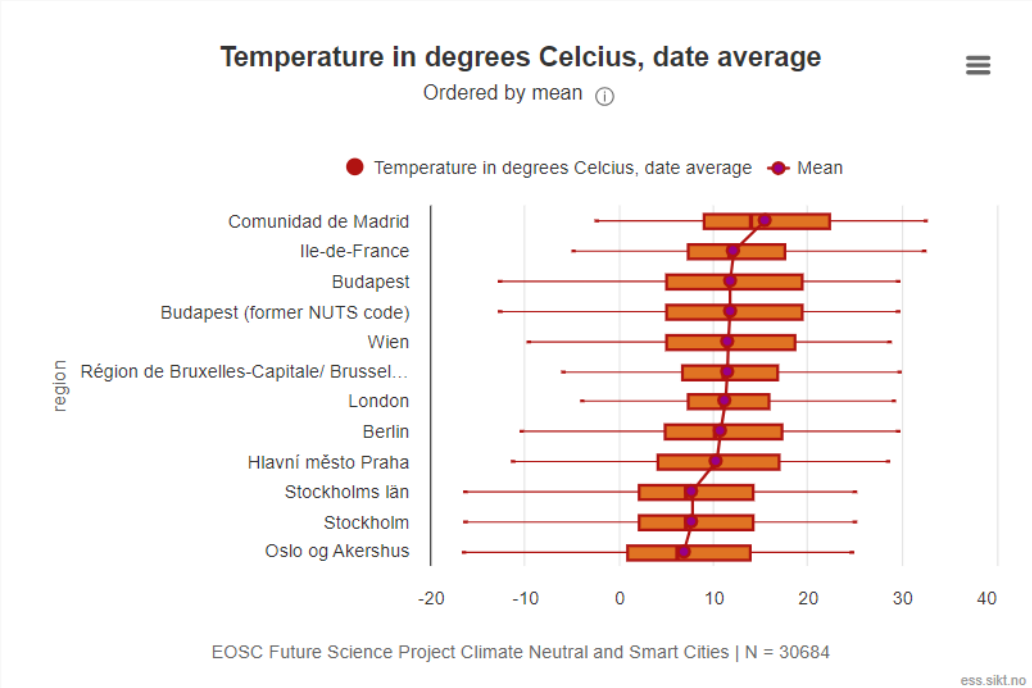
This step uses a script written in Python3.

Diagram of the Process Step

Hint: *Move the mouse cursor over a parameter to see more information. Click on a parameter or a related step to go to the corresponding page.*

Legend: Input parameter — used by → Process step — produces → Output parameter

tmpdca - Temperature in degrees Celcius, date average



Detailed variable information

tmpdca - Temperature in degrees Celcius, date average

Low	Q1	Median	Q3	High	Mean
-16.5	4.8	10.5	16.8	32.4	10.7

Note:
Regional average daily air temperature at 2m height, for 2016-2022. Unit of measure: °C

Process description ↗