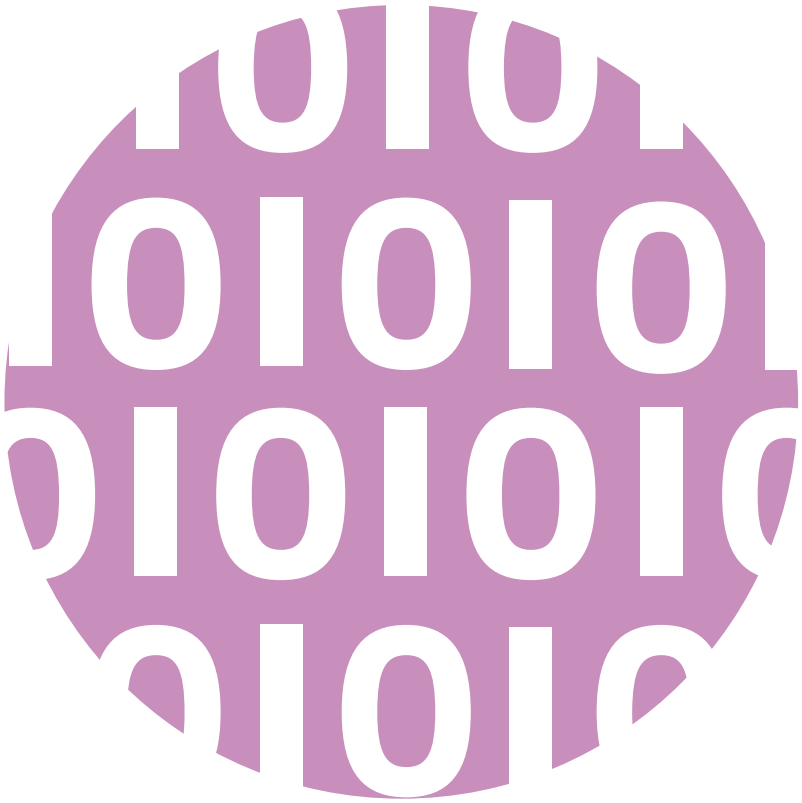
Baltic Marine Environment Protection Commission

BLUES

Co-funded by the European Union

A1.4.2 Annex 2
Model package

2023



HELCOM

BLUES

Co-funded by the European Union

This publication has been produced as part of the project "HELCOM biodiversity, litter, underwater noise and effective regional measures for the Baltic Sea (HELCOM BLUES)". Running from January 2021 to January 2023, HELCOM BLUES is a Helsinki Commission (HELCOM) coordinated project that is co-funded by the European Union. The project is designed to support the third holistic assessment of the ecosystem health of the Baltic Sea (HOLAS 3) as well as the implementation of the HELCOM BSAP. The outcomes can also be used by HELCOM Contracting Parties, that are also EU Member States, to fullfil requirements for their MSFD reporting such as the EU MSFD Initial Assessment and Programmes of Measures (PoMs). Information and views expressed in this publication are the authors' own and might vary from those of the Helsinki Commission or its members.

HELCOM BLUES project website
Baltic Sea Action Plan 2021 (BSAP)
HOLAS 3

This publication is a deliverable of the HELCOM BLUES project's activity 1 - effectiveness and measures.

For bibliographic purposes this document should be cited as: A1.4.2 Annex 2_Model package. HELCOM BLUES (2023).

Contibutors: Antti Iho, Antti-Jussi Kieloaho

Layout: Laura Ramos Tirado

General information about the HELCOM BLUES project

EU programme concerned
Marine Strategy Framework Directive: Support to the preparation of the next 6-year cycle of implementation

Reference number of the call for proposals
DG ENV/MSFD 2020 call

Title of the project
HELCOM biodiversity, litter, underwater noise and effective regional measures for the Baltic Sea (HELCOM BLUES)

Grant agreement number
110661/2020/839624/SUB/ENV.C.2

Name of beneficiary of grant agreement
Baltic Marine Environment Commission – Helsinki Commission (HELCOM)

Official legal form
Intergovernmental Organisation

Official registration number
Not applicable

Official address
Katajanokanlaituri 6B, 00160 Helsinki, Finland

Name and title of the Project Coordinator
Jana Wolf, Project Coordinator

Name and title of the project manager
Jannica Haldin, Deputy Executive Secretary

Name of partners in the project and abbreviations used
Center for Environmental Policy (AAPC)
Kiel Institute for the World Economy (IfW)
Latvian Institute of Aquatic Ecology (LIAE/LHEI)
Natural Resources Institute Finland (LUKE)
Swedish University of Agricultural Sciences (SLU)
Swedish Meteorological and Hydrological Institute (SMHI)
Stockholm University (SU)
Swedish Agency for Marine and Water Management (SwAM/HaV)
Finnish Environment Institute (SYKE)
Tallinn University of Technology (TalTech)
University of Veterinary Medicine Hannover (TiHo)
Center for Earth System Research and Sustainability, University of Hamburg (UHAM-CEN)
University of Tartu (UT)

Sub-contractors
AKTiiVS Ltd (AKTiiVS)
International Council for the Exploration of the Sea (ICES)
Gavia EcoResearch (GAR)
Quiet-Oceans (QO)
Meereszoologie (MZ)
Keep Sweden Tidy (KST/HSR)
Swedish Natural History Museum (NRM)

Start date and end date of the project
25/01/2021 – 24/01/2023

HELCOM_BLUES

Author: Antti-Jussi Kieloaho (antti-jussi.kieloaho@luke.fi; Natural Resources Institute Finland)

Project contains source codes of two separate programs, helcom-api and som. The first program is api consumer to retrieve and process spatial data in raster or vector form. The second set of source code is the first part of a program that processes, organises and calculates data for 'Sufficiency of Measures' analysis done at HELCOM.

Installation

You can use helcom_api or som packages like set of libraries and call them, e.g., by using the jupyter notebooks provided (`src/helcom_api/sandbox.ipynb` or `src/som/sandbox.ipynb`) in Visual Studio Code IDE. You can install correct dependencies in virtual environment defined in `pyproject.toml` :

Python version has to be >= 3.9!

```
cd /path/to/helcom_blues
python -m venv .
source bin/activate
```

If you wish to have editable version of code use:

```
python -m pip install -e .
```

If you want to use run only version:

```
python -m pip install .
```

helcom_api

Contains HELCOM API consumer. Workflow of processing of spatial data fetched from HELCOM API is shown in Fig 1.

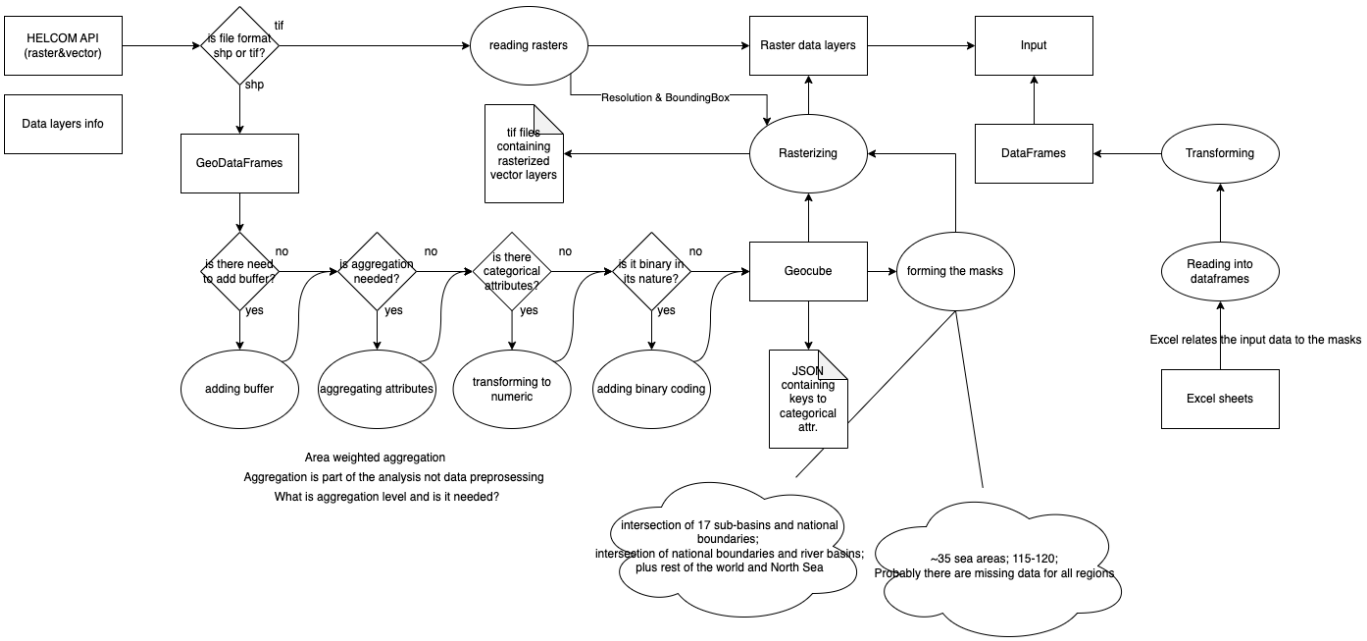


Figure 1. Workflow of processing spatial data.

In `src:helcom_api:data` is directory `EEZ` containing topology corrected spatial shape files for exclusice economic zones and administrative boundaries. These are not available in HELCOM API, but given upon request.

In `configuration.py` there is meta data of spatial data and other information needed in processing.

In `gis_tools.py` contains io and processing functions of workflow.

som (Sufficiency of Measures) model framework

In the first part development of SOM calculation platform, two of three of input data files are used for initialize and populate object model by programatically describing problem domain, in other words, interactions between societal, economic and envrionment in its geographcical context at Baltic Sea region.

In construction of the the model three principles are followed when ever possible: 1. modularity, 2. object with single responsibility, and 3. clear interfaces. Even though, current version is a proof of concept of object model, care has been taken that it can extended in future versions, e.g., expected values extracted from survey data and description of the problem domain are single values with a range (min and max), not distributions, but implementation of distributions is already planned in future versions.

Input data consist three files generalInput.xlsx, measureEffInput.xslx, and pressStateInput.

generalInput.xlsx contains describtion of the problem domain: - in sheet:ActMeas are governance issues, all rows are independent 1. ID - ID of the case 2. MT*ID - Measure type ID 3. In*Activities - Relevant Activities, 0 means all relevant activities for Measure type in sheet:MT*to*A*to*P 4. In*Pressure - Relevant Pressures, 0 means all relevant pressures for Measure type in sheet:MT*to*A*to*P 5. In*State*components - Relevant States, 0 means all relevant states for Measure type in sheet:MT*to*A*to*P 6. Multiplier*OL - Measure type multiplier 7. B*ID - Basin IDs 8. C*ID - Country IDs

- in sheet:ActPres are environmental issues, lists the relevant basins for each Activity-Pressure pairs
  1. Activity - Activity ID
  2. Pressure - Pressure ID
  3. Basin - Basin ID
  4. Ml# - MostLikely (ActivityPressure.AP_expected)
  5. Min# - Minimum end of range
  6. Max# - Maximum end of range
- in sheet:MT*to*A*to*P are linkages between measure types, activities, pressures and states
- measureEffInput.xslx: survey data on the effects of measures on activity-pressure pairs as surved by expert panels
- pressStateInput.xlsx

Model file structure (under `src:som`):

- `__main__` : calls facade functions that represents workflow steps.
- `configuration.py` : holds paths to input data files and other meta data about input files.
- `som_app.py` : contains facade function that wraps underlying functions on thematic collections representing workflow steps.
- `som_classes.py` : contains object definitions of core and secondary object layers.
- `som_tools.py` : contains data reading and processing functions.
- `sandbox.ipynb` : jupyter notebook to run, test and check model objects.

Object model of SOM

The model framework encapsulates problem domain and survey data as a layered structure as shown in Fig 2.

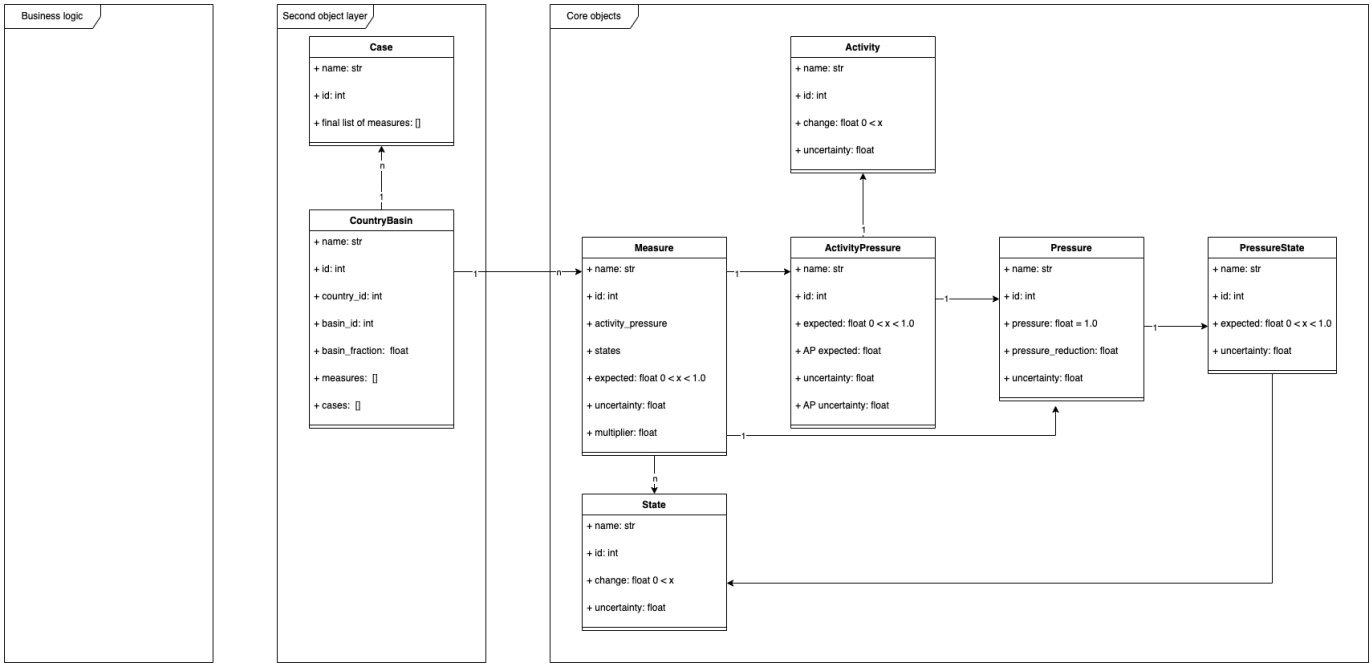


Figure 2. The class diagram of SOM

The layers consists of set of objects that describes basic elements of problem domain and their interactions. Each object contains appropriate data either from description of the problem domain, calculated from survey data, or calculated from all the previous.

Object model describes the core functionality of SOM analysis, object oriented data structure resulted from survey data of expert panel together with description of

problem domain. It concentrates to model Measure, Activity-Pressure, Activity and Pressure interactions and store approriate data. State, Pressure, and State-Pressure are included in object model but not implemented.

In the initialization of object model following steps are taken

1. Process survey data from measure effect input and read general input
2. Build core object model and initialize core object instances
3. Build second object layer model and initialize seconf object layer instances
4. Post-process core objects organised in secondary object layer by storing data into them from general input

The data flow from input data to data structures containing object model is shown in Fig 3. Data flow diagram shows processes and data structures that are created upon a initalization.

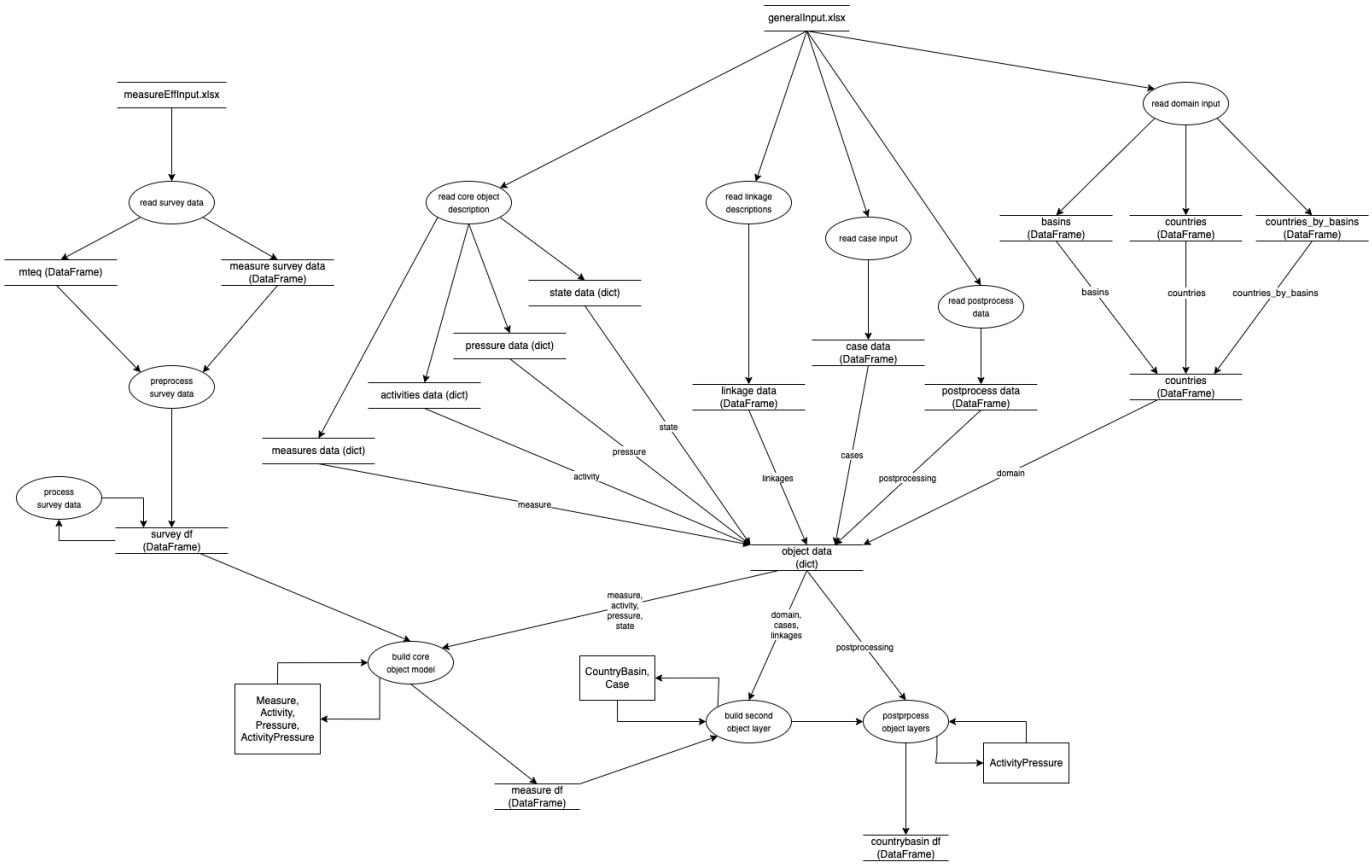


Figure 3. The data flow diagram.

After the initialization, altogether 41 object models are instantiated. They represents 41 country-basin combinations. They are stored in a form of a Pands DataFrame named `countrybasin_df` together with their country-basin, basin and country ids. Each country-basin pair are their own sub-model with a separate deep copies of instantiated core model components. Country-basin pair represents a geographical aspect of SOM analysis. Sub-division to smaller geographical regions is possible, but omitted from this version to keep it simple.

Next development steps

- reading and processing in pressStateInput.xlsx
- implementation of model part combining interactions and data: Pressure, State, Pressure-State
- implementation of simple version of business logic
- module that implements distribution based calculation of business logic
- `helcom_blues/som_tools.py:159: RuntimeWarning: divide by zero encountered in true_divide scaling_factor = np.divide(max_expecte` warning can be easily fixed by adding 'from np.finfo import eps' and adding eps to max_effectivness