Climate Index Metadata and its Implementation

Klaus Zimmermann    Lars Bärring

Rossby Centre
Sveriges Meteorologiska och Hydrologiska Institut

2020-04-07 EGU Sharing Geosciences Online
Climate Index Metadata

Why Metadata?
- Enable automation and interoperability
- Improve understandability and user experience

State of the Art
- CF Conventions & CMIP Data Request

Extend for Climate Index Domain
- YAML files, generated from
- Excel spreadsheet

https://bitbucket.org/cf-index-meta/cf-index-meta
An Implementation: Climix

Another One?

- Existing: icclim, climpact2, xclim, climdex.pcic, ...
- Metadata & calculation details only in code

Features

- Index definition from metadata standard
- All metadata & calculation details from open standard
- Extensible via python entry_points
- Iris & Dask: shared & distributed memory parallelization

https://git.smhi.se/climix/climix
Climate Index Metadata

Outline

Climate Index Metadata
Metadata Examples
Contents

Climix
CF Index Meta

- Aspires to be an emerging community standard
- Current status at
  https://bitbucket.org/cf-index-meta/cf-index-meta
CF Index Meta

Format
- Main document: `master_table.xls`
- Transformed into set of yaml files

Contents
Three parts of metadata
- Index definitions
- Input variable definitions
- Index functions
An Example Index

ttx:
  reference: ETCCDI
  period:
    allowed:
      annual:
      seasonal:
      monthly:
        default: annual
  output:
    var_name: "txx"
    standard_name: air_temperature
    long_name: "Maximum daily maximum temperature"
    units: "degree_Celsius"
    cell_methods:
      - time: maximum within days
      - time: maximum over days

input:
  data: tasmax

index_function:
  name: statistics
  parameters:
    reducer:
      kind: reducer
      reducer: max

ET:
  short_name: "txx"
  long_name: "Maximum daily maximum temperature"
  definition: "Maximum value of daily TX"
**An Example Index**

```plaintext
txx:
  reference: ETCCDI
  period:
    allowed:
      annual:
      seasonal:
      monthly:
        default: annual
  output:
    var_name: "txx"
    standard_name: air_temperature
    long_name: "Maximum daily maximum temperature"
    units: "degree_Celsius"
    cell_methods:
      - time: maximum within days
      - time: maximum over days
  input:
    data: tasmax
  index_function:
    name: statistics
    parameters:
      reducer:
        kind: reducer
        reducer: max
  ET:
    short_name: "txx"
    long_name: "Maximum daily maximum temperature"
    definition: "Maximum value of daily TX"
```
An Example Index

txx:
  reference: ETCCDI
  period:
    allowed:
      annual:
      seasonal:
      monthly:
        default: annual
  output:
    var_name: "txx"
    standard_name: air_temperature
    long_name: "Maximum daily maximum temperature"
    units: "degree_Celsius"
    cell_methods:
      - time: maximum within days
      - time: maximum over days

input:
  data: tasmax

index_function:
  name: statistics
  parameters:
    reducer:
      kind: reducer
      reducer: max

ET:
  short_name: "txx"
  long_name: "Maximum daily maximum temperature"
  definition: "Maximum value of daily TX"
An Example Index

txx:
  reference: ETCCDI
  period:
    allowed:
    annual:
    seasonal:
    monthly:
    default: annual
output:
  var_name: "txx"
  standard_name: air_temperature
  long_name: "Maximum daily maximum temperature"
  units: "degree_Celsius"
  cell_methods:
    - time: maximum within days
    - time: maximum over days
input:
  data: tasmax
index_function:
  name: statistics
  parameters:
    reducer:
      kind: reducer
      reducer: max
ET:
  short_name: "txx"
  long_name: "Maximum daily maximum temperature"
  definition: "Maximum value of daily TX"
An Example Index

txx:
    reference: ETCCDI
    period:
        allowed:
            annual:
            seasonal:
            monthly:
                default: annual
    output:
        var_name: "txx"
        standard_name: air_temperature
        long_name: "Maximum daily maximum temperature"
        units: "degree_Celsius"
        cell_methods:
            time: maximum within days
            time: maximum over days
    input:
        data: tasmax
    index_function:
        name: statistics
        parameters:
            reducer:
                kind: reducer
                reducer: max

ET:
    short_name: "txx"
    long_name: "Maximum daily maximum temperature"
    definition: "Maximum value of daily TX"
Climate Index Metadata

An Example Index

```plaintext
taxx:
  reference: ETCCDI
  period:
    allowed:
      annual:
      seasonal:
      monthly:
    default: annual
  output:
    var_name: "txx"
    standard_name: air_temperature
    long_name: "Maximum daily maximum temperature"
    units: "degree_Celsius"
    cell_methods:
      - time: maximum within days
      - time: maximum over days

input:
  data: tasmax

index_function:
  name: statistics
  parameters:
    reducer:
      kind: reducer
      reducer: max

ET:
  short_name: "txx"
  long_name: "Maximum daily maximum temperature"
  definition: "Maximum value of daily TX"
```

input data to be operated on refers to variable metadata (see below)
An Example Index

txx:
  reference: ETCCDI
  period:
    allowed:
      annual:
      seasonal:
      monthly:
    default: annual
  output:
    var_name: "txx"
    standard_name: air_temperature
    long_name: "Maximum daily maximum temperature"
    units: "degree_Celsius"
    cell_methods:
      - time: maximum within days
      - time: maximum over days
  input:
    data: tasmax
  index_function:
    name: statistics
    parameters:
      reducer:
        kind: reducer
        reducer: max
  ET:
    short_name: "txx"
    long_name: "Maximum daily maximum temperature"
    definition: "Maximum value of daily TX"
tasmax:
  standard_name: air_temperature
  cell_methods:
    - time: maximum
  aliases:
    - tasmaxadjust
    - tmax
    - tx
    - maxt
    - TMAX
    - Tmax
    - TX
    - MAXT
    - maxT
spell_length:
  description: | [Climate Index Metadata]
  Calculates statistics on lengths of spells.
  First, the threshold is transformed to the same standard_name and units as
  the input data.
  Then the thresholding is performed as condition(data, threshold),
  ie if condition is <, data < threshold.
  Then the spells are determined, and finally the statistics according to
  the specified reducer are calculated.

parameters:
  threshold:
    kind: quantity
  condition:
    kind: operator
  reducer:
    kind: reducer
## Contents

<table>
<thead>
<tr>
<th>Reference</th>
<th>Number of indices</th>
<th>ready</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ETCCDI</td>
<td>27</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>ET-SCI</td>
<td>33</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>ECA&amp;D</td>
<td>44</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>CLIPC</td>
<td>52</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>B4EST</td>
<td>2</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>SMHI</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>5</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td><strong>total</strong></td>
<td><strong>164</strong></td>
<td><strong>100</strong></td>
<td></td>
</tr>
</tbody>
</table>

- **ready**: all metadata in place
- Number of index functions: 12
Outline

Climate Index Metadata
Metadata Examples
Contents

Climix
Overview

- Python package to calculate climate indices
- Index definition directly from metadata
- Based on Iris and Dask
- Version 1.0.0 expected by July 2020
- Technology preview at
  https://git.smhi.se/climix/climix
Index Definitions

- Directly read from metadata yaml files
- Add new, custom indices by adding yaml files; no coding needed
Architecture I

Climix...

- Implements index functions
- Sets up Dask environment
- Reads input data
- Stores output
Architecture II

**entry_points for Index Functions**

- Connect metadata and implementation
- Easily add new or alternative implementations

```python
setuptools.setup(
    entry_points={
        'climix.index_functions': [
            'spell_length=climix.index_functions:SpellLength',
        ],
    },
)```
Summary

CF Index Meta
- Emerging community standard
- Build on CF and extend it

Climix
- Open Source, Python climate index package
- Strongly metadata oriented
Acknowledgment

This work is supported by
- the European project IS-ENES3
- SMHI Rossby Centre