02 Transformation Module - EOL in 2025

What	*nameofinstance*.xml
Description	Configuration for the transformation module
schema location	https://fewsdocs.deltares.nl/schemas/version1.0/transformationSets.xsd

Entry in ModuleDescriptors	`<moduleDescriptor id="Transformation">`
	`  <description>General Transformation Component</description>`
	`  <className>nl.wldelft.fews.system.plugin.transformation.TransformationController</className>`
	`</moduleDescriptor>`

⊙ This module has reached its end-of-life (EOL) status. This means that there will be no active development of this module. Configurators are urged to use the new transformation module instead, see Transformation Module (Improved schema).

Transformation Module Configuration

The Transformation module is a general-purpose module that allows for generic transformation and manipulation of time series data. The module may be configured to provide for simple arithmetic manipulation, time interval transformation, shifting the series in time etc, as well as for applying specific hydro-meteorological transformation such as stage discharge relationships etc.

The Transformation module allows for the manipulation and transformation of one or more time series. The utility may be configured to provide for;

- Manipulation of one or more series using a standard library of arithmetic operators/functions (enumerated);
  - Addition, subtraction, division, multiplication
  - Power function, exponential function
- Hydro-meteorological functions like:
  - Deriving discharges from stages
  - Compute potential evaporation
  - Calculating weighted catchment average rainfall
- Shifting series in time
- Time interval conversion:
  - Aggregation
  - Dis-aggregation
  - Converting non-equidistant to equidistant series
- Creating astronomical tide series from harmonic components
- Handling of typical profiles
- Data hierarchy
- Selection of (tidal) peaks
- statistics

When available as configuration on the file system, the name of the XML file for configuring an instance of the transformation module called for example TransformHBV_Inputs may be:

TransformHBV_Inputs 1.00 default.xml

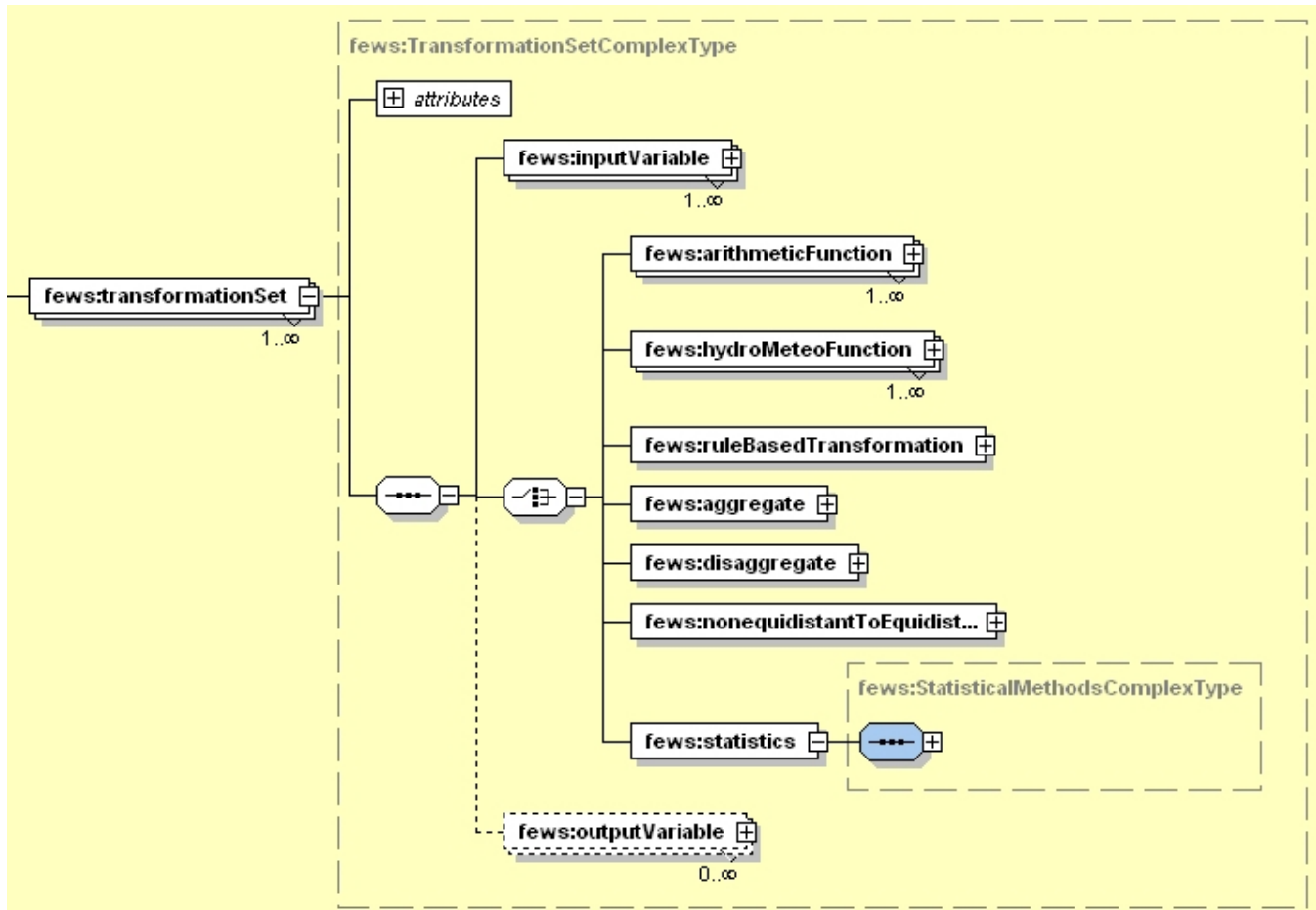| TransformHBV_Inputs	File name for the TransformHBV_Inputs configuration.
1.00	Version number
default	Flag to indicate the version is the default configuration (otherwise omitted).

Figure 57 Root element of the Transformation module.

transformationSet

Root element for the definition of a transformation (processing an input to an output). Multiple entries may exist.

Attributes;

- **transofrmationId** : Id of the transformation defined. Used for reference purposes only. This Id will be included in log messages generated
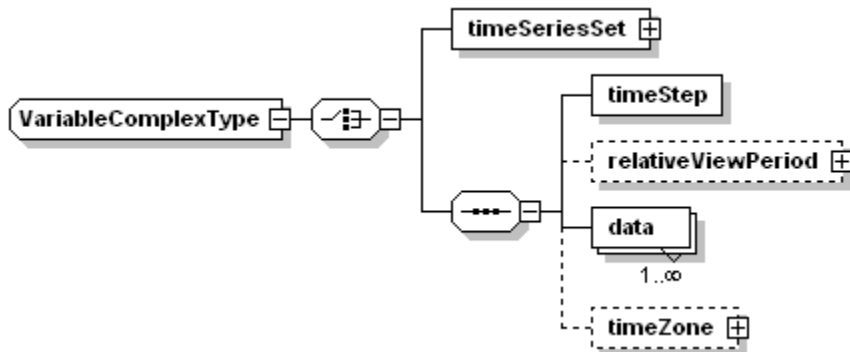

Figure 58 Elements of the definition of an input variable.

inputVariable

Definition of the input variables to be used in transformation. This may either be a time series set, a typical profile or a set of (harmonic) components. The InputVariable is assigned an ID. This ID is used later in the transformation functions as a reference to the data.

Attributes;

- **variableId** : ID of the variable (group).Later used in referencing the variable.

- **variableType** : Optional type definition of variable (defaults to "any")
- **convertDatum** : Optional Boolean flag to indicate if datum is to be converted. Note that all input variables should specify the same value for convertDatum, otherwise a warning is given for input variables for which a datum applies.

Available harmonic components are listed in the attached file.

timeSerieSet

Definition of an input variable as a time series set (see TimeSeriesSet definition).

timeStep

Time step for typical profile if variable to be defined as typical profile.

Attributes;

- **unit** (enumeration of: second, minute, hour, day, week, nonequidistant)
- **multiplier** defines the number of units given above in a time step (not relevant for nonequidistant time steps)**
- **divider** same function as the multiplier, but defines fraction of units in time step.**

relativeViewPeriod

Relative view period of the typical profile to create. If this is defined and the time span indicated is longer than the typical profile data provided, then the profile data will be repeated until the required time span is filled. If the optional element is not provided then the typical profile data will be used only once.

data

Data entered to define the typical profile. Data can be entered in different ways. The typical profile can be defined as a series of values at the requested time step, inserted at the start of the series, or it can be mapped to specific time values (e.g. setting a profile value to hold at 03:15 of every day). Which of these is used depends on the attributes defined.

Attributes;

- **value** : Required value for each step in the profile
- **monthDay** : Attribute value indicating the value entered is valid for a month/day combination. The year value is added depending on the year value in which it is used. The string has the format "-[month][day]". For example the 23rd of August is "--08-23".
- **dateTime** : Attribute value indicating the value entered is valid for a specific date time combination. The string has the format "[year][month][day]T[hour]:[minute]:[second]". For example the 23rd of August is "1984-12-31T00:00:00".
- **time** : Attribute value indicating the value entered is valid for a specific time, irrespective of the date. The date value is added run time. The string has the format "[hour]:[minute]:[second]". For example "01:15:00".

timeZone

Optional specification of the time zone for the data entered (see timeZone specification).

timeZone:timeZoneOffset

The offset of the time zone with reference to UTC (equivalent to GMT). Entries should define the number of hours (or fraction of hours) offset. (e.g. +01:00)

timeZone:timeZoneName

Enumeration of supported time zones. See appendix B for list of supported time zones.

arithmeticFunction

Root element for defining a transformation as an arithmetic function (see next section for details).

hydroMeteoFunction

Root element for defining one of the available hydro-meteorological transformations.

ruleBasedTransformation

Root element for defining a rule based transformation (see next section for details on rules).
Attributes;

- **rule** : definition of aggregation approach. Enumeration of;
- *selectpeakvalues*
- *selectlowvalues*
- *selectpeakvalueswithincertaingap*
- *selectlowvalueswithincertaingap*
- *equitononequidistant*
- *equitononequidistantforinstantaneousseries*
- *equitononequidistantforaccumulativeseries*

- *datahierarchy*
- *typicalprofiletotimeseries*
- *zerodegreealtitudelevel*
- *datatotimeseries*

aggregate

Root element for defining a time aggregation transformation (rules are discussed below)
Attributes;

- **rule** : definition of aggregation approach. Enumeration of;
- *instantaneous*
- *accumulative*
- *mean*
- *constant*

disaggregate

Root element for defining a time dis-aggregation transformation (rules are discussed below)
Attributes;

- rule: definition of disaggregation approach. Enumeration of;
- *instantaneous*
- *accumulative*
- *disaggregateusingweights*
- *constant*

nonequidistantToEquidistant

Root element for defining transformation of an non-equidistant time series to an equidistant time series. (rules are discussed below)
Attributes;

- rule: definition of approach. Enumeration of;
- *zero*
- *missing*
- *linearinterpolated*
- *equaltolast*

Statistics

Root element for defining statistical transformations.
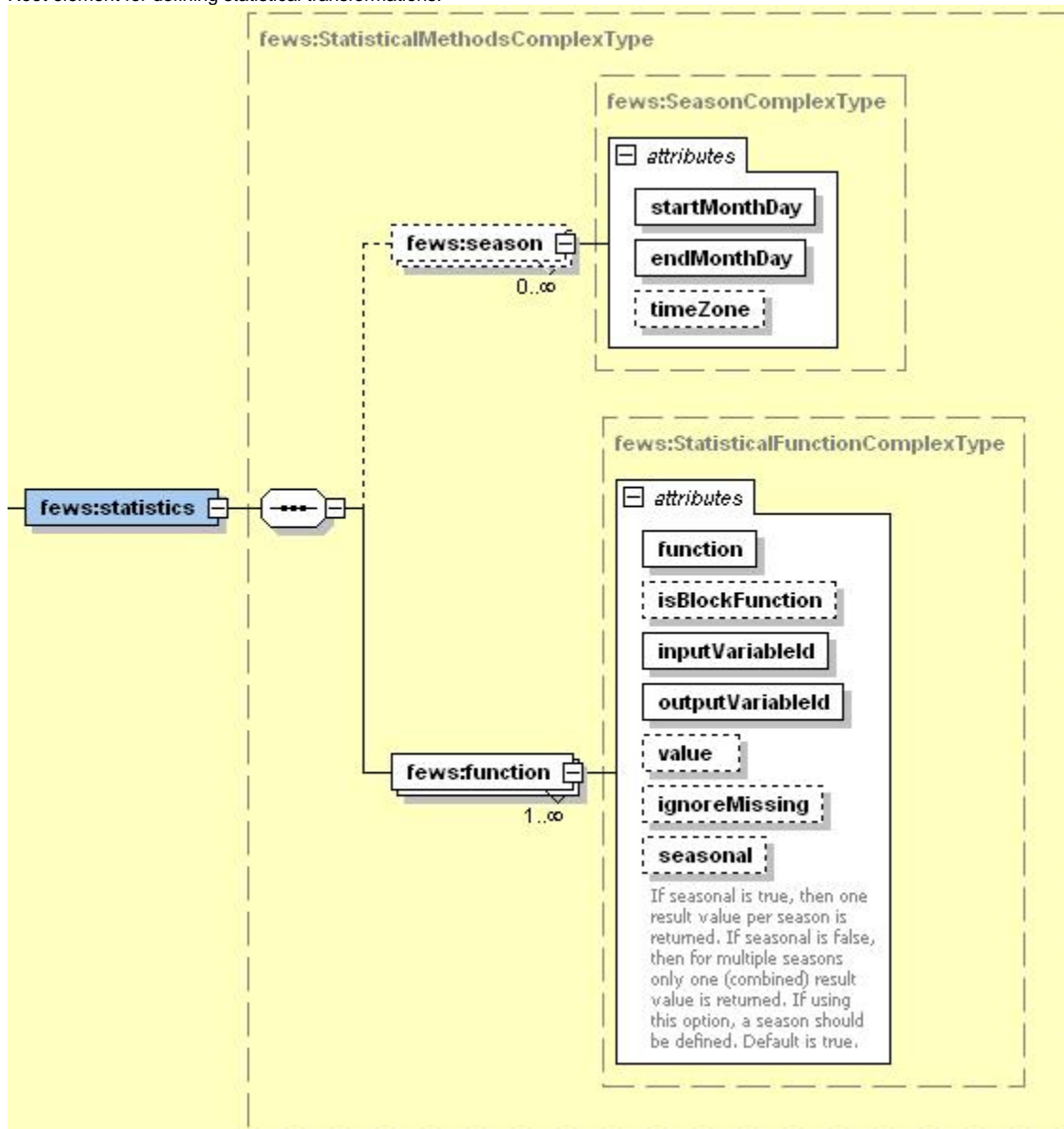
Season: the statistics transformation can also be carried out for a specific season which is defined by a start and end date. If multiple seasons are specified, then the statistics transformation will be carried out separately for each specified season. A warning will be given when seasons overlap in time.

- **startMonthDay:** defines start time of season "--mm-dd"
- **endMonthDay:** defines end time of season "--mm-dd"
- **timeZone**

Function:

- **available functions** *
    - max
    - min
    - sum
    - count
    - mean
    - median
    - standardDeviation
    - percentileExceedence
    - percentileNonExceedence
    - quartile
    - skewness
    - kurtosis
    - variance
    - rsquared
    - rootMeanSquareError

- **isBlockFunction:** *if true, the statistical parameters are calculated for each time window defined by the time step of the output time series, e.g. time step year leads to yearly statistical parameters. If false and output time series time step is set to nonequidistant, the statistical parameters are calculated for the relative view period (one value for the whole period) or for the individual season if applied.
- **inputVariableId**
- **outputVariableId**
- **value:** if function percentileExceedence or percentileNonExceedence is chosen, the desired percentile has to be defined, e.g. 75-th percentile => value="75"
- **ignoreMissing:** if true, all missings of the input time series are not taken into account in the statistical calculation.
- **seasonal:** this option is only relevant when using seasons. If true (default), then one result value per season per year is returned. If false, then for each season only one (combined) result value is returned. For example when seasonal is false, the month January is specified as a season, the input time series contains data for a period of ten years and the function max is specified, then the result will be the maximum of all values in January in all ten years. Note: if a specific season (e.g. January 2006) is not fully contained within the input time series, then this specific season is not used in the calculations. For example if the month January is specified as a season and the input time series contains only data from 15 January 2006 to 1 March 2008, then only January 2007 and January 2008 will be used in the calculations. In this case January 2006 will not be used in the calculations.

ArithmeticFunction & hydroMeteoFunction

Through definition of an arithmetic function, a user defined equation can be applied in transforming a set of input data to a set of output data. Any number of inputs may be defined, and used in the user defined function. Each input variable is identified by its Id, as this is used configuring the function. The function is written using general mathematical operators. A function parser is used in evaluating the functions (per time step) and returning results. These are again assigned to variables which can be linked to output time series through the variableId.

Rather than use a usedDefinedFunction, a special function can also be selected from a list of predefined hydroMeteoFunctions. When selected this will pose requirements on other settings.

Transformations may be applied in segments, with different functions or different parameters used for each segment. A segment is defined as being valid for a range of values, identified in one of the input variables (see example below).
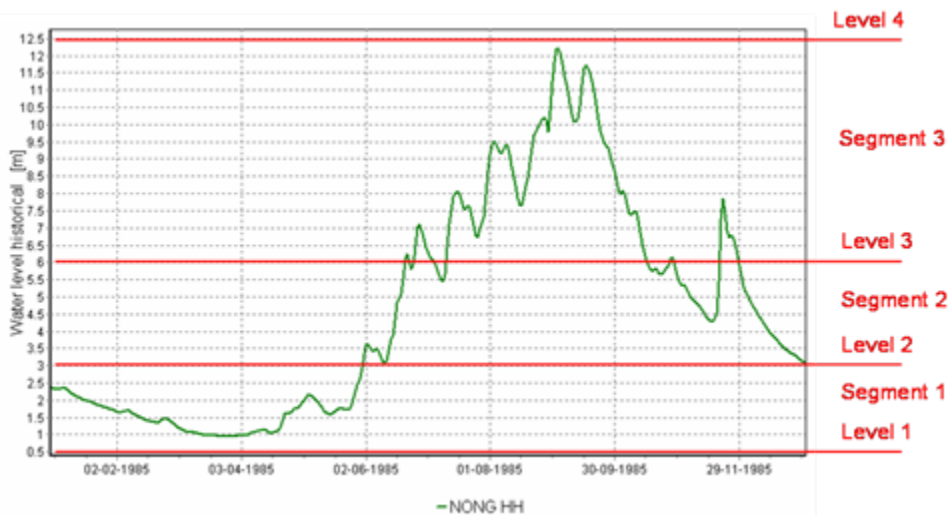

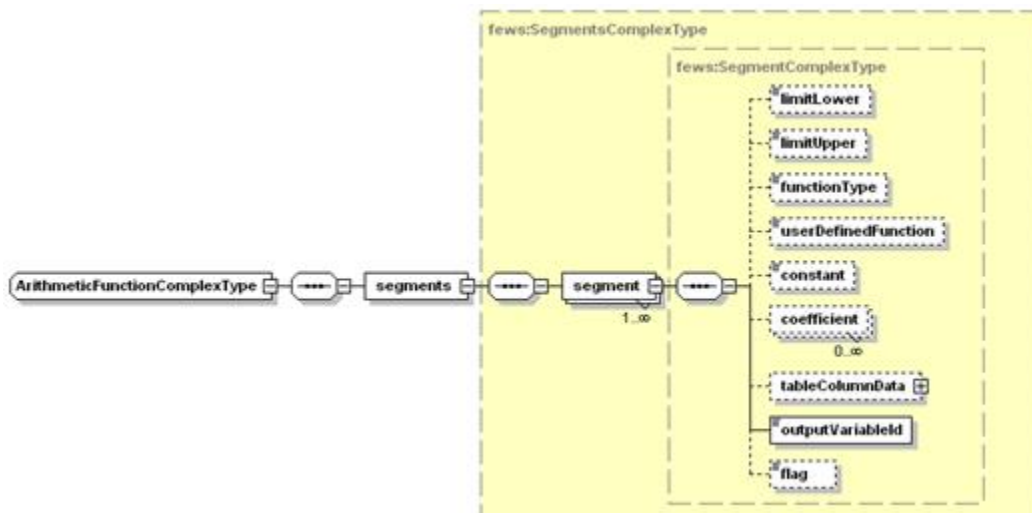Figure 59 Example of applying segments to a time series

Figure 60 Elements of the Arithmetic section of the transformation module configuration

segments

Root element for defining segments. When used this must include the input variable Id used to determine segments as an attribute.
Attributes;

- **limitVariablId** : Id of input variable used to test against segment limits.

segment
Root element for definition of a segment. At least one segment must be included.

limitLower
Lower limit of the segment. Function defined will be applied at a given time step only if value at that time step in the variable defined as limitVariable is above or equal to this value.

limitUpper
Upper limit of the segment. Function defined will be applied at a given time step only if value at that time step in the variable defined as limitVariable is below this value (below or equal only for the highest segment).

functionType

Element used only when defining a predefined hydroMeteoFunction. Depending on selected function, specific requirements will hold for defining input variables and parameters. If a special function is selected then the user defined function element is not defined; Enumeration of available options is (the most important are discussed below);

- *simpleratingcurve* ; for applying a simple power law rating curve.
- *weigthtedaverage* : special function for calculating weighted average of inputs. When a value in one of the inputs is missing, the remaining inputs will be used and the weights rescaled to unity.
- *penman:* for calculating evaporation using Penman
- *penmannortheast:* specific implementation of Penman formula
- *qhrelationtable* : allows application of a rating curve using a table.
- *degreemanipulation*
- *accumulation*: this calculates a moving sum. For this the window needs to be configured. For a given output time the output value equals the sum of the input values within the period (currentOutputTime - window, currentOutputTime). The start of the period is exclusive and the end of the period is inclusive.

userDefinedFunction

Optional specification of a user defined function to be evaluated using the function parser. Only the function need be defined, without the equality sign. The function is defined as a string and may contain Id's of inputSeries, names of variables and constants defined, and mathematical operators
Operators offered

- scalar series: +, -, /, *, ^, sin, cos, tan, asin, acos, atan, sinh, cosh, tanh, asinh, acosh, atanh, log, ln, exp, sqrt, abs, pow, min, max, minSkipMissings, maxSkipMissings, sumSkipMissings, average
- operators for conversion of grid to scalar series: spatialMin, spatialMax, spatialSum, spatialSumSkipMissings, spatialAverage

h54 constant
Allows definition of a constant to be used in the function.

coefficient

Optional element to allow coefficients for use in the function to be defined. These coefficients are allocated and Id for later use in the function defined. For user defined functions specific coefficients need to be defined. Multiple entries may be defined.
Attributes;

- **coefficientId** : Id of the coefficient defined. This can be used in the function.
- **coefficientType** : identification of the coefficient. Applied in rule based configurations.
- **value:** value of the coefficient

tableColumnData

Definition of a table to use for transforming input variable to output variables.
Attributes;

- nrOfColumns: number of columns in table (should equal 2).
- variableIdCol1 Input variable associated with first column
- variableIdCol2 Output variable associated with first column

tableColumnData:data

Element containing data for each row in the table
Attributes;

- col1: value for column 1
- col2: value for column 2

outputVariable

Id of the output variable from the function. This may be saved to the database by associating the Id to an outputVariable.

Optional element to force saving the result data for the segment with a given flag. This may be used for example to force data from a segment as doubtful. Enumeration is either "unreliable" or "doubtful". if data is reliable the element should not be included.

Stage-Discharge and Discharge-Stage transformation

Stage discharge transformations can be defined using the *simpleratingcurve* option of the hydroMeteoFunctions. To apply this certain properties must be defined in each segment.

For stage-discharge transformation the requirements are;

- Coefficient values for coefficientId's "a", "b" and "c" must be defined.
- Rating curve formula is $Q = a * (H+b) \wedge c$
- Input variable Id must be "H"
- Output variable Id must be "Q".
- limitVariableId must be "H".

Example:

For stage-discharge transformation the requirements are;

- Coefficient values for coefficientId's "a", "b" and "c" must be defined.
- Input variable Id must be "Q"
- Output variable Id must be "H".
- limitVariableId must be "Q".

Example:

```
<hydroMeteoFunction>
        <segments limitVariableId="Q">
                <segment>
                        <limitLower>0</limitLower>
                        <limitUpper>15.53</limitUpper>
                        <functionType>simpleratingcurve</functionType>
                        <coefficient coefficientId="a" value="30.6187"/>
                        <coefficient coefficientId="b" value="0.065"/>
                        <coefficient coefficientId="c" value="1.9997"/>
                        <outputVariableId>H</outputVariableId>
                </segment>
                <segment>
                        <limitLower>15.53</limitLower>
                        <limitUpper>9999</limitUpper>
                        <functionType>simpleratingcurve</functionType>
                        <coefficient coefficientId="a" value="44.3036"/>
                        <coefficient coefficientId="b" value="-0.145"/>
                        <coefficient coefficientId="c" value="1.5205"/>
                        <outputVariableId>H</outputVariableId>
                </segment>
        </segments>
</hydroMeteoFunction>
```

Establishing catchment average precipitation

Catchment average rainfall can be determined by weighting input precipitation time series. The *weightedaverage* option of the hydroMeteoFunctions can be applied to include the option of recalculation of weights if one of the input locations is missing. To apply this certain properties must be defined in each segment.

For establishing catchment average precipitation the requirements are;

- functionType must be set to *weightedaverage*
- Weights are given as coefficient values with coefficientId's "a", "b" and "c" etc.
- Additional coefficients may be defined to allow for altitude correction.

Example:

Aggregation, disaggregation and non-equidistant to equidistant

This set of transformations allows temporal aggregation and disaggregation of time series. The time step defined in the input variable and the output variable determine the howthe time steps are migrated. The configuration need only define the rule followed in aggregation/disaggregation. Aggregation and disaggregation can only be used to transform between equidistant time steps. A nonequidistant series can be transformed to an equidistant series using the appropriate element (see above).

Aggregation rules;

- *Instantaneous:* apply instantaneous resampling- ie value at cardinal time step in output series is same as in input time series at that time step.
- *accumulative* : value in output time series is accumulated sum of values of time steps in input time series (use in for example aggregating rainfall in mm).
- *mean* value in output time series is mean of values of time steps in input time series (use in for example aggregating rain rate in mm/hr).

- *constant*

Disaggregation rules;

- *Instantaneous:* apply linear interpolation- ie value at cardinal time step in output series is same as in input time series at that time step. Values in between are interpolated.
- *accumulative* : value in output time series is derived as equal fraction of valuein input series. Fraction is determined using ration of time steps.
- *Disaggregateusingweights* value in output time series weighted fraction of input value. Weights are defined as coefficients. These are sub-elements to the disaggregation element. The number of coefficients defined should be equal to the disaggregation ration (i.e. 24 when disaggregating from day to hour). The coefficient Id's should be numbered 1 to n..
- *constant* value in output time series at intermediate time steps is equal to the last available value in the input time series.
  Rules for mapping non-equidistant time series to equidistant time series
- *zero* value in output time series is zero if time values do not coincide
- *missing* value in output time series is missing if time values do not coincide
- *linearinterpolated* value in output time series is interpolated linearly between neighbouring values in input time series
- *equaltolast* value in output time series is equal to last available value in input time series.

Rule based transformations

The set of rule based transformations is a library of specific data transformation functions. Configuration of the rule based transformation is the same as in the Arithmetic transformation. However, each rule may have specific requirements on the elements that need to be defined. Many parameters that affect the transformation will need to be defined as a coefficient, using the appropriate coefficientType definition.

The rule based transformations can be grouped into four main sections;

- Selection of peak or low values from a time series.
- Resampling a equidistant time series set using time values specified in a non-equidistant time series set.
- Data hierarchy
- Various transformations.

Selection of peak or low flow values

Selection of peaks and lows

Set of rules to allow selection of peaks and lows from an input time series.

Enumerations in the rule attribute of the ruleBasedTransformation element;

- *selectpeakvalues*
- *selectlowvalues*
- *selectpeakvalueswithincertaingap*
- *selectlowvalueswithincertaingap*
  ——
  The first two enumerations will select all peaks or lows in the time series. The second two will select peaks only if there is a defined gap in time between peaks. If not they are considered to be of dependent and only the highest peak of the dependent sets will be returned.

Requirements for definitions of peak selections using gaps to define independence are;

- A coefficientId "a" must be defined. The coefficientType must be set to "gaplengthinsec". The value attribute defines the length of the minimum gap in seconds.
- A coefficientId "b" must be defined with coefficientType "peaksbeforetimezero". The value attribute defines the maximum number of peaks to consider before T0.
- A coefficientId "c" must be defined with coefficientType "peaksaftertimezero". The value attribute defines the maximum number of peaks to consider before T0.
- A coefficientId "d" must be defined with coefficientType "totalnumberofpeaks". The value must be set to zero.

The following two coefficients are optional:

- A coefficientId "e" with coefficientType "skipjustbeforetimezero" indicates how many peaks to skip just before T0.
- A coefficientId "f" with coefficientType "skipjustaftertimezero" indicates how many peaks to skip just after T0.

They default to 0.

Example:

```
<ruleBasedTransformation rule="selectpeakvalueswithincertaingap">
    <segments limitVariableId="X1">
        <segment>
            <coefficient coefficientId="a" coefficientType="gaplengthinsec" value="2700"/>
            <coefficient coefficientId="b" coefficientType="peaksbeforetimezero" value="3"/>
            <coefficient coefficientId="c" coefficientType="peaksaftertimezero" value="4"/>
            <coefficient coefficientId="d" coefficientType="totalnumberofpeaks" value="0"/>
            <coefficient coefficientId="e" coefficientType="skipjustbeforetimezero" value="2"/>
            <coefficient coefficientId="f" coefficientType="skipjustaftertimezero" value="2"/>
            <outputVariableId>Y1</outputVariableId>
        </segment>
    </segments>
</ruleBasedTransformation>
```

In this example:

- The time between two local maxima (peaks) should be at least 2700 seconds or 45 minutes.
- Only the last three peaks before T0 and the first four peaks after T0 are considered.
- The last two peaks just before T0 are skipped, leaving only the third last one.
- Similarly the first peaks just after T0 are skipped, leaving the third and fourth ones.

Sampling values from equidistant time series

This section of the rule based transformation can be applied to sample items from an equidistant time series at the time values in a non-equidistant time series. This may be required when applying transformations to a non-equidistant time series. The values to add will first need to be resampled to the right time value. An example is when wind and wave information is required at the time of the tidal peaks for entry in a lookup table.

Enumerations in the rule attribute of the ruleBasedTransformation element;

- *equitononequidistant*
- *equitononequidistantforinstantaneousseries*
- *equitononequidistantforaccumulativeseries*

 —
 The first two elements are equivalent. The last will consider accumulations of the input variable up to the time value sampled.

Requirements for definitions of resampling equidistant time series are;

- The limitVariableId attribute of the segements element must be the non-equidistant time series which determines the time values at which the equidistant series is to be sampled.
- The userDefinedFunction must contain the equidistant time series to be sampled
- The outputVariableId must resolve to a non-equidistant time series.

 —
 Example:

```
<ruleBasedTransformation
                rule="equitononequidistantforinstantaneousseries">
      <segments limitVariableId="Hastro">
            <segment>
                  <userDefinedFunction>Hlevel</userDefinedFunction>
                  <outputVariableId>Hpeaks</outputVariableId>
            </segment>
      </segments>
</ruleBasedTransformation>
```

Data Hierarchy

This is a simple method to merge overlapping equidistant time series in a single equidistant series. Gaps in foremost (first) series will be filled with data of second series if a valid value is available at the current time step, otherwise the gap is filled with data from the third series and so on until no more time series are available. Only missing data values and unreliable values are filled. Doubtful values remain in the result series as doubtful.
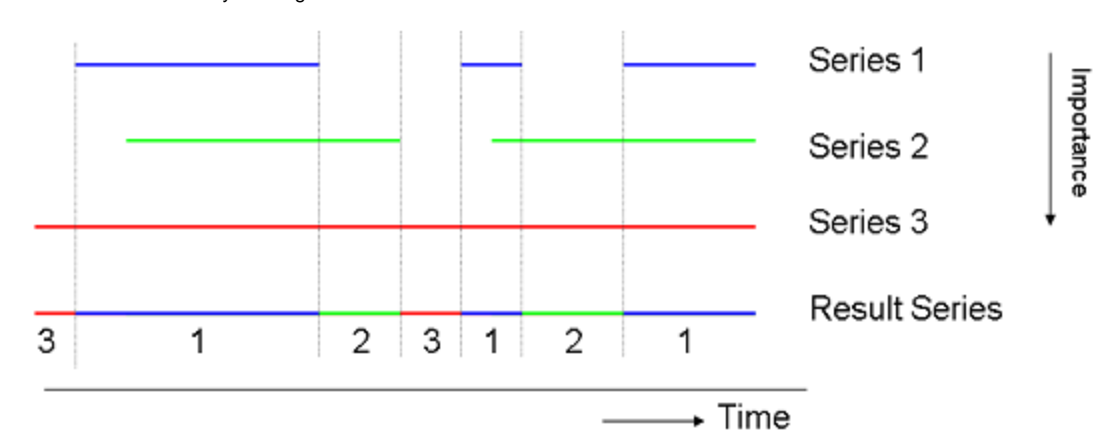

Figure 61 Schematic example of merging series using data hierarchy.

In example above Series 1 is the most important time series, Series 2 has a lower hierarchy and series 3 has the lowest hierarchy. The resulting time series has values from all 3 series as shown in figure above.

Data hierarchy poses no specific requirements to variables defined. Only the Id of the output variable is of importance.

Creating time series from typical profiles

Typical profiles can be defined in the inputVariable as described above. To use a typical profile it must first be mapped to a dynamic time series. This can then be retrieved in a later configuration of a module for use.

Enumerations in the rule attribute of the ruleBasedTransformation element;

- *typicalprofiletotimeseries:*
- *datatotimeseries*

The first type of mapping is used when the typical profile has a concept of date/time (e.g. must be mapped to specific dates or time values). The second is used when only a series of data is given. The time series is then filled with the first data element given as the first time step of the relative view period to be created.

Typical profile mapping poses no specific requirements to variables defined. Only the Id of the output variable is of importance.

Max Gap Length

For linear tranformations, the maxGapLength rule can be used. This way, interpolation will only occur if values are apart by no more then the defined gap length.

When a time is defined for which a value should be interpolated, the TimeSpan between the two surrounding values (the one before and the one after) is compared to the defined maxGapLength TimeSpan. If the maxGapLength TimeSpan is equal or larger, a value for that time will be interpolated. In the other case, the value will not be calculated or set to missing.

outputVariable

Definition of the output variables to be written following transformation. See the inputVariable for the attributes and structure. The output variable can only be a TimeSeriesSet (typical profiles are only used as inputs). The OutputVariable is assigned an ID. This ID must be defined as the result of the transformation.