

---

# J2K

Dokumentation

Peter Krause

JAMS version: 3.0\_b24 (20150705)

Copyright © 2016

22.02.2016 17:28:39

## Zusammenfassung

The standard J2K hydrological model

## Inhaltsverzeichnis

Modelleinfuehrung .....	4
Modellstruktur .....	4
ArrayToScalar .....	9
Metainformationen .....	9
Attribute .....	9
CalcAreaWeight .....	9
Metainformationen .....	9
Attribute .....	10
Beschreibung der Komponente .....	10
CalcDailyNetRadiation .....	10
Metainformationen .....	10
Attribute .....	10
Beschreibung der Komponente .....	11
CalcDailySolarRadiation .....	12
Metainformationen .....	12
Attribute .....	12
Beschreibung der Komponente .....	13
CalcExtraterrRadiation .....	13
Metainformationen .....	13
Attribute .....	13
Beschreibung der Komponente .....	14
CalcLanduseStateVars .....	15
Metainformationen .....	15
Attribute .....	15
Beschreibung der Komponente .....	16
CalcLanduseStateVars_trop .....	16
Metainformationen .....	16
Attribute .....	17
CalcLatLong .....	17
Metainformationen .....	17
Attribute .....	17
Beschreibung der Komponente .....	18
CalcRainSnowParts .....	18
Metainformationen .....	18
Attribute .....	19
Beschreibung der Komponente .....	19

---

CalcRelativeHumidity .....	20
Metainformationen .....	20
Attribute .....	20
Beschreibung der Komponente .....	21
DoubleConditionalContext .....	21
Metainformationen .....	21
Attribute .....	21
DoubleSetter .....	21
Metainformationen .....	22
Attribute .....	22
Beschreibung der Komponente .....	22
DoubleTransfer .....	22
Metainformationen .....	22
Attribute .....	22
GeomReader .....	23
Metainformationen .....	23
Attribute .....	23
Beschreibung der Komponente .....	23
IDWWeightCalculator .....	23
Metainformationen .....	24
Attribute .....	24
Beschreibung der Komponente .....	24
InitJ2KProcessGroundwater .....	25
Metainformationen .....	25
Attribute .....	25
Beschreibung der Komponente .....	25
InitJ2KProcessLumpedSoilWaterStates .....	26
Metainformationen .....	26
Attribute .....	26
Beschreibung der Komponente .....	27
J2KArrayGrabber .....	27
Metainformationen .....	27
Attribute .....	27
J2KMultiplication .....	28
Metainformationen .....	28
Attribute .....	28
J2KProcessGroundwater .....	29
Metainformationen .....	29
Attribute .....	29
Beschreibung der Komponente .....	30
J2KProcessInterception .....	32
Metainformationen .....	32
Attribute .....	32
Beschreibung der Komponente .....	33
J2KProcessLumpedSoilWater_DPSmod .....	34
Metainformationen .....	34
Attribute .....	34
J2KProcessReachRouting .....	36
Metainformationen .....	36
Attribute .....	36
Beschreibung der Komponente .....	38
J2KProcessSnow .....	39
Metainformationen .....	39
Attribute .....	39

---

---

Beschreibung der Komponente .....	40
JAMSTContext .....	43
Metainformationen .....	43
Attribute .....	44
Beschreibung der Komponente .....	44
JAMSTExecInfo .....	44
Metainformationen .....	44
Attribute .....	44
Beschreibung der Komponente .....	44
JAMSTModel .....	44
Metainformationen .....	45
Attribute .....	45
Beschreibung der Komponente .....	45
JAMSTSpatialContext .....	45
Metainformationen .....	45
Attribute .....	45
Beschreibung der Komponente .....	46
JAMSTTemporalContext .....	46
Metainformationen .....	46
Attribute .....	46
Beschreibung der Komponente .....	47
JAMSTUnitConverter .....	47
Metainformationen .....	47
Attribute .....	47
MapCreator .....	48
Metainformationen .....	48
Attribute .....	48
Beschreibung der Komponente .....	49
MapCreator3D .....	49
Metainformationen .....	49
Attribute .....	49
Beschreibung der Komponente .....	50
PenmanMonteith .....	50
Metainformationen .....	50
Attribute .....	50
Beschreibung der Komponente .....	51
RainCorrectionRichter .....	54
Metainformationen .....	54
Attribute .....	54
Beschreibung der Komponente .....	55
Regionalisation .....	56
Metainformationen .....	56
Attribute .....	56
Beschreibung der Komponente .....	57
StandardEfficiencyCalculator .....	59
Metainformationen .....	59
Attribute .....	59
Beschreibung der Komponente .....	60
StandardEntityReaderUpstreamTopo .....	62
Metainformationen .....	62
Attribute .....	63
StandardGroundwaterParaReader .....	63
Metainformationen .....	63
Attribute .....	64

---

Beschreibung der Komponente .....	64
StandardLUReader .....	64
Metainformationen .....	64
Attribute .....	65
Beschreibung der Komponente .....	65
StandardSoilParaReader .....	65
Metainformationen .....	65
Attribute .....	66
Beschreibung der Komponente .....	66
SubbasinFlooding .....	66
Metainformationen .....	66
Attribute .....	67
SumAggregator .....	67
Metainformationen .....	67
Attribute .....	67
Beschreibung der Komponente .....	68
TSDataStoreReader .....	68
Metainformationen .....	68
Attribute .....	68
Beschreibung der Komponente .....	69
TSPlot .....	69
Metainformationen .....	70
Attribute .....	70
Beschreibung der Komponente .....	71
TemporalSumAggregator .....	72
Metainformationen .....	72
Attribute .....	73
Beschreibung der Komponente .....	73
VariableAdder .....	73
Metainformationen .....	73
Attribute .....	74
WeightedSumAggregator .....	74
Metainformationen .....	74
Attribute .....	74
Beschreibung der Komponente .....	74
Modellparameter .....	75
Literaturverzeichnis .....	83

## Modelleinfuehrung

Bitte "documentation/template\_de.xml" im Workspace anpassen, um diesen Text zu aendern!

## Modellstruktur

Modell Name= *J2K*

- Kontextkomponente Klasse= *jams.model.JAMSTextContext* Name= *ParameterInput*
  - Komponente Klasse= *io.StandardEntityReaderUpstreamTopo* Name= *EntityReader*
  - Komponente Klasse= *org.unijena.j2k.io.StandardLUReader* Name= *LUReader*
  - Komponente Klasse= *org.unijena.j2k.io.StandardSoilParaReader* Name= *STReader*

- Komponente Klasse= *org.unijena.j2k.io.StandardGroundwaterParaReader* Name= *GWReader*  
end Kontextkomponente *ParameterInput*
- Kontextkomponente Klasse= *jams.model.JAMSSpatialContext* Name= *Initialization*
- Kontextkomponente Klasse= *jams.model.JAMSSpatialContext* Name= *CatchmentInit*
  - Komponente Klasse= *org.unijena.j2k.aggregate.SumAggregator* Name= *AreaCalculator*  
end Kontextkomponente *CatchmentInit*
- Kontextkomponente Klasse= *jams.model.JAMSSpatialContext* Name= *HRUInit*
  - Komponente Klasse= *org.unijena.j2k.CalcAreaWeight* Name= *AreaWeight*
  - Komponente Klasse= *org.unijena.j2k.geographicalCalculations.CalcLatLong* Name= *CalcLatLong*
  - Komponente Klasse= *org.unijena.j2k.inputData.CalcLanduseStateVars\_trop* Name= *CalcLanduseStateVars\_trop*
  - Komponente Klasse= *org.unijena.j2k.inputData.CalcLanduseStateVars* Name= *CalcLanduseStateVars*
  - Komponente Klasse= *org.unijena.j2k.radiation.CalcExtraterrRadiation* Name= *ExtRad*
  - Komponente Klasse= *org.unijena.j2k.soilWater.InitJ2KProcessLumpedSoilWaterStates* Name= *InitSoilWater*
  - Komponente Klasse= *org.unijena.j2k.groundwater.InitJ2KProcessGroundwater* Name= *InitGroundWater*
  - Komponente Klasse= *org.unijena.j2k.regionalisation.IDWWeightCalculator* Name= *tminWeights*
  - Komponente Klasse= *org.unijena.j2k.regionalisation.IDWWeightCalculator* Name= *tmeanWeights*
  - Komponente Klasse= *org.unijena.j2k.regionalisation.IDWWeightCalculator* Name= *tmaxWeights*
  - Komponente Klasse= *org.unijena.j2k.regionalisation.IDWWeightCalculator* Name= *ahumWeights*
  - Komponente Klasse= *org.unijena.j2k.regionalisation.IDWWeightCalculator* Name= *sunhWeights*
  - Komponente Klasse= *org.unijena.j2k.regionalisation.IDWWeightCalculator* Name= *windWeights*
  - Komponente Klasse= *org.unijena.j2k.regionalisation.IDWWeightCalculator* Name= *raincorrWeights*
  - Komponente Klasse= *org.unijena.j2k.regionalisation.IDWWeightCalculator* Name= *precipWeights*
  - Komponente Klasse= *org.unijena.j2k.DoubleSetter* Name= *ZeroSetter*end Kontextkomponente *HRUInit*

end Kontextkomponente *Initialization*

- Kontextkomponente Klasse= *jams.model.JAMSTemporalContext* Name= *TimeLoop*
- Komponente Klasse= *jams.components.gui.JAMSExecInfo* Name= *ExecutionInfo*
- Komponente Klasse= *org.unijena.j2k.DoubleSetter* Name= *CatchmentResetter*
- Kontextkomponente Klasse= *jams.model.JAMSTContext* Name= *TSInput*
  - Komponente Klasse= *jams.components.io.TSDataStoreReader* Name= *TmeanDataReader*
  - Komponente Klasse= *jams.components.io.TSDataStoreReader* Name= *TminDataReader*
  - Komponente Klasse= *jams.components.io.TSDataStoreReader* Name= *TmaxDataReader*
  - Komponente Klasse= *jams.components.io.TSDataStoreReader* Name= *AhumDataReader*
  - Komponente Klasse= *jams.components.io.TSDataStoreReader* Name= *RaincorrDataReader*
  - Komponente Klasse= *jams.components.io.TSDataStoreReader* Name= *PrecipDataReader*
  - Komponente Klasse= *jams.components.io.TSDataStoreReader* Name= *SunhDataReader*
  - Komponente Klasse= *org.unijena.j2k.inputData.RainCorrectionRichter* Name= *RCRichter*
  - Komponente Klasse= *jams.components.io.TSDataStoreReader* Name= *WindDataReader*
  - Komponente Klasse= *jams.components.io.TSDataStoreReader* Name= *ObsRunoffDataReader*

end Kontextkomponente *TSInput*

- Kontextkomponente Klasse= *jams.model.JAMSSpatialContext* Name= *HRULoop*
- Kontextkomponente Klasse= *jams.model.JAMSTContext* Name= *Regionalization*
  - Komponente Klasse= *org.unijena.j2k.J2KArrayGrabber* Name= *j2kArrayGrabber*
  - Komponente Klasse= *org.unijena.j2k.regionalisation.Regionalisation* Name= *TmeanRegionaliser*
  - Komponente Klasse= *org.unijena.j2k.regionalisation.Regionalisation* Name= *TminRegionaliser*
  - Komponente Klasse= *org.unijena.j2k.regionalisation.Regionalisation* Name= *TmaxRegionaliser*
  - Komponente Klasse= *org.unijena.j2k.regionalisation.Regionalisation* Name= *AhumRegionaliser*
  - Komponente Klasse= *org.unijena.j2k.regionalisation.Regionalisation* Name= *RaincorrRegionaliser*
  - Komponente Klasse= *org.unijena.j2k.regionalisation.Regionalisation* Name= *PrecipRegionaliser*

- Komponente Klasse= *org.unijena.j2k.regionalisation.Regionalisation* Name= *SunhRegionaliser*
- Komponente Klasse= *org.unijena.j2k.regionalisation.Regionalisation* Name= *WindRegionaliser*
- end Kontextkomponente *Regionalization*
- Komponente Klasse= *org.unijena.j2k.VariableAdder* Name= *VariableAdder*
- Komponente Klasse= *org.unijena.j2k.VariableAdder* Name= *VariableAdder\_1*
- Komponente Klasse= *org.unijena.j2k.DoubleSetter* Name= *HRUResetter*
- Komponente Klasse= *org.unijena.j2k.inputData.CalcRelativeHumidity* Name= *CalcRelativeHumidity*
- Komponente Klasse= *org.unijena.j2k.mathematicalCalculations.J2KMultiplication* Name= *J2KMultiplication*
- Komponente Klasse= *org.unijena.j2k.radiation.CalcDailySolarRadiation* Name= *CalcDailySolarRadiation*
- Komponente Klasse= *org.unijena.j2k.radiation.CalcDailyNetRadiation* Name= *CalcDailyNetRadiation*
- Komponente Klasse= *org.unijena.j2k.potET.PenmanMonteith* Name= *CalcPotentialET*
- Komponente Klasse= *org.unijena.j2k.snow.CalcRainSnowParts* Name= *CalcRainSnowParts*
- Komponente Klasse= *org.unijena.j2k.interception.J2KProcessInterception* Name= *J2KProcessInterception*
- Komponente Klasse= *org.unijena.j2k.snow.J2KProcessSnow* Name= *J2KProcessSnow*
- Kontextkomponente Klasse= *jams.components.conditional.DoubleConditionalContext* Name= *DoubleConditionalContext*
  - Komponente Klasse= *soilWater.J2KProcessLumpedSoilWater\_DPSmod* Name= *J2KProcessLumpedSoilWater*
  - Komponente Klasse= *soilWater.J2KProcessLumpedSoilWater\_DPSmod* Name= *J2KProcessLumpedSoilWater\_flood*
- end Kontextkomponente *DoubleConditionalContext*
- Komponente Klasse= *org.unijena.j2k.groundwater.J2KProcessGroundwater* Name= *J2KProcessGroundwater*
- Komponente Klasse= *jams.components.aggregate.TemporalSumAggregator* Name= *TemporalSumAggregator1*
- Komponente Klasse= *jams.components.datatransfer.DoubleTransfer* Name= *HRU2HRURouting*
- Komponente Klasse= *jams.components.datatransfer.DoubleTransfer* Name= *HRU2ReachRouting*

- Komponente Klasse= *org.unijena.j2k.aggregate.WeightedSumAggregator* Name=  
*SpatialWeightedSumAggregator1*
- Komponente Klasse= *org.unijena.j2k.aggregate.WeightedSumAggregator* Name=  
*SpatialWeightedSumAggregator2*
- end Kontextkomponente *HRULoop*
- Kontextkomponente Klasse= *jams.model.JAMSSpatialContext* Name= *ReachLoop*
- Komponente Klasse= *org.unijena.j2k.DoubleSetter* Name= *ZeroSetter\_1*
- Komponente Klasse= *org.unijena.j2k.routing.J2KProcessReachRouting* Name=  
*J2KProcessReachRouting*
- Komponente Klasse= *org.unijena.j2k.aggregate.SumAggregator* Name=  
*ChannelStorageAggregator*
- Komponente Klasse= *flooding.SubbasinFlooding* Name= *SubbasinFlooding*
- end Kontextkomponente *ReachLoop*
- Komponente Klasse= *org.unijena.j2k.aggregate.WeightedSumAggregator* Name=  
*FlowWeightedSumAggregator*
- Komponente Klasse= *jams.components.tools.JAMSUnitConverter* Name= *SimRunoffConverter*
- Komponente Klasse= *org.unijena.j2k.ArrayToScalar* Name= *ObsRunoff*
- Kontextkomponente Klasse= *jams.model.JAMSContext* Name= *TSVisualization*
- Komponente Klasse= *jams.components.gui.TSPlot* Name= *Runoff (overall)*
- Komponente Klasse= *jams.components.gui.TSPlot* Name= *Runoff (components)*
- Komponente Klasse= *jams.components.gui.TSPlot* Name= *Soil moisture rel*
- Komponente Klasse= *jams.components.gui.TSPlot* Name= *Soil moisture act*
- Komponente Klasse= *jams.components.gui.TSPlot* Name= *ETPlot*
- Komponente Klasse= *jams.components.gui.TSPlot* Name= *Radiation*
- Komponente Klasse= *jams.components.gui.TSPlot* Name= *Temp*
- Komponente Klasse= *jams.components.gui.TSPlot* Name= *Rhum*
- Komponente Klasse= *jams.components.gui.TSPlot* Name= *Wind*
- Komponente Klasse= *jams.components.gui.TSPlot* Name= *Percolation*
- Komponente Klasse= *jams.components.gui.TSPlot* Name= *RG2*
- Komponente Klasse= *jams.components.gui.TSPlot* Name= *Storages*
- Komponente Klasse= *jams.components.gui.TSPlot* Name= *Snow*



```

    end Kontextkomponente TSVisualization

    • Komponente Klasse= org.unijena.j2k.efficiencies.StandardEfficiencyCalculator Name= Efficiencies

    end Kontextkomponente TimeLoop

    • Kontextkomponente Klasse= jams.model.JAMSTContext Name= MapVisualization

    • Komponente Klasse= jams.components.io.GeomReader Name= GeometryReader

    • Komponente Klasse= jams.components.gui.MapCreator Name= Map

    • Komponente Klasse= jams.components.gui.MapCreator3D Name= Map3D

    end Kontextkomponente MapVisualization

end Modell J2K

```

## ArrayToScalar

### Metainformationen

Klassifikation	
Paket	org.unijena.j2k
Autor	Sven Kralisch
Version	1.0_1
Datum	2013-09-13

### Attribute

#### Eingabe

Name	Beschreibung	Einheit	Typ
dataArray	array of values to extract from		DoubleArray
arrayIndex	the index of the value to be extracted		Integer[]
factor	optional correction factor		Double

#### Ausgabe

Name	Beschreibung	Einheit	Typ
dataValue	extracted value		Double[]

## CalcAreaWeight

Eine Komponente zur Bestimmung des relativen Flächengewichtes

### Metainformationen

Klassifikation	
----------------	--

Paket	org.unijena.j2k
Autor	Peter Krause
Version	1.0_0
Datum	2011-05-30

## Attribute

### Eingabe

Name	Beschreibung	Einheit	Typ
entityArea	die Fläche der Modelleinheit	m <sup>2</sup>	Double
catchmentArea	die Gesamtfläche des Einzugsgebiets	m <sup>2</sup>	Double

### Ausgabe

Name	Beschreibung	Einheit	Typ
areaWeight	das relative Flächengewicht der Modelleinheit	n/a	Double

## Beschreibung der Komponente

Diese Komponente bestimmt das relative Flächengewicht einer Modelleinheit indem die Fläche der Modelleinheit durch die Gesamtfläche dividiert wird. Dieses Gewicht ist für flächengewichtete Summen oder Mittelwertbildungen von Bedeutung.

## CalcDailyNetRadiation

Eine Komponenten zur Berechnung der täglichen Nettostrahlung in MJ/m<sup>2</sup>/d als Eingangsgröße für die Verdunstungsberechnung. Die Berechnung erfolgt weitestgehend nach .

## Metainformationen

Klassifikation	
Paket	org.unijena.j2k.radiation
Autor	Peter Krause
Version	1.0_0
Datum	2011-05-30

## Attribute

### Eingabe

Name	Beschreibung	Einheit	Typ
tmean	Die mittlere Lufttemperatur [°C]	°C	Double
rhum	Die relative Luftfeuchtigkeit [%]	%	Double

Name	Beschreibung	Einheit	Typ
extRad	Die extraterrestrische Einstrahlung [MJ/m <sup>2</sup> /d]	MJ m <sup>-2</sup> d <sup>-1</sup>	Double
solRad	Die Globalstrahlung [MJ/m <sup>2</sup> /d]	MJ m <sup>-2</sup> d <sup>-1</sup>	Double
albedo	Die Albedo der Landbedeckung [-]		Double
elevation	Die Höhe über NN der räumlichen Modelleinheit [m]	m	Double

## Ausgabe

Name	Beschreibung	Einheit	Typ
netRad	Die Nettostrahlung [MJ/m <sup>2</sup> /d]	MJ m <sup>-2</sup> d <sup>-1</sup>	Double
swRad	Der kurzwellige Strahlungsanteil [MJ/m <sup>2</sup> /d]	MJ m <sup>-2</sup> d <sup>-1</sup>	Double
lwRad	Der langwellig Strahlungsanteil [MJ/m <sup>2</sup> /d]	MJ m <sup>-2</sup> d <sup>-1</sup>	Double
refETNetRad	tägliche Nettostrahlung für die Grasreferenzverdunstung [MJ/m <sup>2</sup> /d]	MJ m <sup>-2</sup> d <sup>-1</sup>	Double

## Beschreibung der Komponente

Die Nettostrahlung dient als Energiequelle für Verdunstungsberechnungen. Sie ergibt sich aus der extraterrestrischen Einstrahlung, die beim Durchgang durch die Atmosphäre zur Globalstrahlung reduziert wird. Durch Abzug der langwelligen Strahlung von der Globalstrahlung entsteht die Nettostrahlung. Für die Berechnung sind folgende Einzelschritte notwendig:

Berechnung des **Sättigungsdampfdrucks** ( $es$ ) in kPa bei Lufttemperatur ( $T_{avg}$ ):

$$es = 0.6108 \times e^{\frac{17.27 \times T_{avg}}{273.3 + T_{avg}}} \text{ [kPa]}$$

Daraus ergibt sich mit Hilfe der relativen Luftfeuchtigkeit der **aktuelle Dampfdruck** ( $ea$ ) nach:

$$ea = es \times \frac{U}{100} \text{ [kPa]}$$

Aus der extraterrestrischen Strahlung ( $R_a$ ) und der Höhe  $h$  der Modelleinheit ergibt sich die **Globalstrahlung bei unbedecktem, klarem Himmel** ( $R_{CS}$ ):

$$R_{CS} = (0.75 - 2 \times 10^{-5} \times h) \times R_a \text{ [MJ/m}^2\text{/d]}$$

Aus Globalstrahlung ( $R_G$ ) und der Albedo ( $\alpha$ ) der jeweiligen Landbedeckung ergibt sich die **Nettokurzwellenstrahlung** ( $R_{SW}$ ):

$$R_{SW} = (1 - \alpha) \times R_G \text{ [MJ/m}^2\text{/d]}$$

Die **Nettolangwellenstrahlung** ( $R_{LW}$ ) berechnet sich aus der Boltzmannkonstante ( $= 4.903 \times 10^{-9}$  [MJ/ $K^4$ /m<sup>2</sup>/d]), mittleren absoluten Lufttemperatur ( $T_{abs}$ ) in K, dem aktuellen Dampfdruck ( $ea$ ) in kPa, der Globalstrahlung ( $R_G$ ) und der Globalstrahlung bei unbedecktem Himmel ( $R_{CS}$ ) beide in MJ/m<sup>2</sup>/d nach:

$$R_{LW} = 4.903 \times 10^{-9} \times T_{abs}^4 \times (0.34 - 0.14 \times \sqrt{ea}) \times \left(1.35 \times \frac{R_G}{R_{CS}} - 0.35\right) [\text{MJ/m}^2/\text{d}]$$

Als Differenz aus  $R_{SW}$  und  $R_{LW}$  ergibt sich schließlich die **Nettostrahlung** nach:

$$R_N = R_{SW} - R_{LW} [\text{MJ/m}^2/\text{d}]$$

Ist die langwellige Ausstrahlung größer als die kurzwellige Einstrahlung, wird die Nettostrahlung auf Null gesetzt.

## CalcDailySolarRadiation

Eine Komponente zur Berechnung der Globalstrahlung in täglicher oder monatlicher zeitlicher Auflösung.

## Metainformationen

Klassifikation	
Paket	org.unijena.j2k.radiation
Autor	Peter Krause
Version	1.0_0
Datum	2011-05-30

## Attribute

### Eingabe

Name	Beschreibung	Einheit	Typ
time	Der aktuelle Zeitschritt, das aktuelle Datum		Calendar
sunh	Die täglichen Sonnenscheinstunden [h/d]	h/d	Double
actSIAsCf	Neigungs-Expositions-Korrekturfaktor		Double
latitude	Die geographische Breite der Modelleinheit [deg]	deg	Double
actExtRad	Die tägliche extraterrestrische Strahlung [MJ/m <sup>2</sup> /d]	MJ / m <sup>2</sup> d	Double
angstrom_a	Der Faktor a der Angstrom Gleichung [-] [0..1]		Double
angstrom_b	Der Faktor b der Angstrom Gleichung [-] [0..1]		Double
tempRes	Die zeitliche Auflösung der Modellierung [d   m]		String

### Ausgabe

Name	Beschreibung	Einheit	Typ
sunhmax	Die maximal mögliche Sonnenscheindauer [h]	h/d	Double
solRad	Die tägliche Globalstrahlung [MJ/m <sup>2</sup> /d]	MJ / m <sup>2</sup> d	Double

## Beschreibung der Komponente

Die Globalstrahlung ist der an der Erdoberfläche ankommende Strahlungsanteil. Die Globalstrahlung ergibt sich aus der extraterrestrischen Strahlung abzüglich des im Atmosphärendurchgang reflektierten bzw. adsorbierten Anteils. Dieser kann durch die Angstromformel abgeschätzt werden. Für die Berechnung der Globalstrahlung werden im Modul folgende Einzelberechnungen, entsprechend *Allen et.al 1998* durchgeführt. Für monatliche Berechnungen wird die Globalstrahlung jeweils für den 15. des betreffenden Monats berechnet.

Deklination ( $d$ ) der Sonne für den julianischen Tag ( $jD$ ):

$$d = 0.40954 \times \sin(0.0172 \times (jD - 79.35)) \text{ [rad]}$$

Stundenwinkel bei Sonnenuntergang für die jeweilige geographische Breite ( $lat$  in radians) und die Deklination ( $d$ ):

$$sw = \cos^{-1}(-1 \times \tan lat \times \tan d) \text{ [rad]}$$

Die maximal mögliche Anzahl der Sonnenscheinstunden für den julianischen Tag ( $jD$ ) aus dem Stundenwinkel ( $sw$ ):

$$s_0 = \frac{24}{H} \times sw$$

Mit Hilfe der Angstromformel, die die beiden Parameter  $a$  und  $b$  besitzt, dem Verhältnis zwischen aktuellen ( $s$ ) und maximal möglichen ( $s_0$ ) Sonnenscheinstunden und der extraterrestrischen Strahlung ( $R_A$ ) berechnet sich dann die Globalstrahlung ( $R_G$ ) nach:

$$R_G = \left(a + b \times \frac{s}{s_0}\right) \times R_A \text{ [MJ/m}^2\text{/d]}$$

Zu weiterführenden Informationen über Kontexte siehe auch *Allen et.al 1998*

## CalcExtraterrestrRadiation

Eine Komponente zur Berechnung der extraterrestrischen Einstrahlung.

## Metainformationen

Klassifikation	
Paket	org.unijena.j2k.radiation
Autor	Peter Krause
Version	1.0_0
Datum	2011-05-30

## Attribute

### Eingabe

Name	Beschreibung	Einheit	Typ
latitude	die geographische Breite der jeweiligen Modelleinheit	degree	Double

Name	Beschreibung	Einheit	Typ
longitude	die geographische Länge der jeweiligen Modelleinheit	degree	Double
longTZ	die geographische Bezugslänge der aktuellen Zeitzone	degree	Double
tempRes	die zeitliche Auflösung des Modells	n/a	String
locGrw	die Lage (westlich oder östlich) von Greenwich		String

## Ausgabe

Name	Beschreibung	Einheit	Typ
extRadArray	Ein Array, das die extraterrestrische Einstrahlung für jeden diskreten Zeitschritt (entsprechend der zeitlichen Modellauflösung) enthält.	MJ / m <sup>2</sup> timeUnit	DoubleArray

## Beschreibung der Komponente

Diese Komponente berechnet die extraterrestrische Einstrahlung ( $R_a$ ) für die jeweilige Modelleinheit für ein Standardjahr. Hierfür wird für jeden diskreten Zeitschritt (Stunde, Tag oder Monat)  $R_a$  berechnet und als Arrayelement in der Variablen *extRadArray* abgelegt. Hier kann sie dann später während der Modelllaufzeit abgeholt werden. Die Berechnung der extraterrestrischen Strahlung erfolgt nach [ Allen et.al 1998 ]. Die Berechnung für Monatswerte erfolgt jeweils für einen Tag in der Mitte des Monats und einer Multiplikation des Wertes mit der Anzahl der Tage des betreffenden Monats. Die Berechnung von täglichen Werten erfolgt in folgenden Berechnungsschritten:

Deklination ( $d$ ) der Sonne für den julianischen Tag ( $jD$ ):

$$d = 0.40954 \times \sin(0.0172 \times (jD - 79.35)) [\text{rad}]$$

Solarkonstante ( $sc$ ) für den julianischen Tag ( $jD$ ):

$$sc = 81930 + 2910 \times \cos\left(\frac{\pi}{180} \times (jD - 15)\right) \div 1000000 [\text{MJ/m}^2/\text{min}]$$

Inverse relative Distanz Erde Sonne für den julianischen Tag :  $jD$ ):

$$irdes = 1 + 0.033 \times \cos\left(\frac{2\pi}{365} \times jD\right) [\text{radians}]$$

Stundenwinkel bei Sonnenuntergang für die jeweilige geographische Breite ( $lat$  in radians) und die Deklination  $d$ :

$$sw = \cos^{-1}(-1 \times \tan(lat) \times \tan(d)) [\text{radians}]$$

Extraterrestrische Strahlung  $R_a$ :

$$R_a = \frac{24 \times 60}{\pi} \times sc \times irdes \times (sw \times \sin(lat) \times \sin(d) + \cos(lat) \times \cos(d) \times \sin(sw)) [\text{MJ/m}^2/\text{d}]$$

Für stündliche Werte ergeben sich zusätzliche Berechnungsschritte:

Stundenwinkel in der Mitte der Stunde:

$$msw = \frac{\pi}{12} \times ((midTime + 0.06667 \times (longTz - longitude) + corrS) - 12)$$

wobei *midTime* die Mitte der aktuellen Stunde darstellt und *corrS* ein saisonaler Korrekturfaktor für die Sonnenzeit ist, der sich folgendermaßen berechnet:

$$corrS = 0.1645 \times \sin(2b) - 0.1255 \times \cos(b) - 0.025 \times \sin(b)$$

mit

$$b = \frac{2 \times \pi \times (jD - 81)}{364}$$

Der Stundenwinkel zu Beginn der Stunde :*ssw*):

$$ssw = msw - \left( \pi \times \frac{1}{24} \right)$$

Der Stundenwinkel am Ende der Stunde :*esw*):

$$esw = msw + \left( \pi \times \frac{1}{24} \right)$$

Hiermit ergibt sich die stündliche extraterrestrische Einstrahlung nach:

$$Ra = \frac{12 \times 60}{\pi} \times sc \times irdes \times ((esw - ssw) \times \sin(lat) \times \sin(d) + \cos(lat) \times \cos(d) \times (\sin(esw) - \sin(ssw))) \text{ [MJ/m}^2\text{/h]}$$

Zu weiterführenden Informationen siehe auch: [ Krause 2001 ] [ Allen et.al 1998 ]

## CalcLanduseStateVars

Eine Komponente zur Berechnung der extraterrestrischen Einstrahlung.

## Metainformationen

Klassifikation	
Paket	org.unijena.j2k.inputData
Autor	Peter Krause
Version	1.0_0
Datum	2011-05-30

## Attribute

## Eingabe

Name	Beschreibung	Einheit	Typ
entities	Der Satz der räumlichen Modelleinheiten.		EntityCollection
elevation	Die Höhe ü.N.N. der spezifischen Modelleinheit.	m	Double

Name	Beschreibung	Einheit	Typ
tempRes	Die zeitliche Auflösung der Modellierung - (h) stündlich, (d) täglich, (m) monatlich.		String

## Ausgabe

Name	Beschreibung	Einheit	Typ
LAIArray	Für jede Modelleinheit werden hier die LAI Werte in der zeitlichen Modellauflösung (h, d, m) für ein Standardjahr abgelegt.		DoubleArray
effHArray	Für jede Modelleinheit werden hier die Werte für die effektive Wuchshöhe in der zeitlichen Modellauflösung (h, d, m) für ein Standardjahr abgelegt.		DoubleArray
rsc0Array	Die monatliche Werte für die Stomatawiderstände der Vegetation auf der Modelleinheit.	s / m	DoubleArray

## Beschreibung der Komponente

In dieser Komponente werden Landnutzungsparameter für die entsprechende zeitliche Auflösung des Modells (Variable *tempRes*) und für jede räumliche Modelleinheit berechnet und in entsprechenden Variablen abgelegt. Die Berechnung erfolgt anhand spezifischer Stützwerte, die das Modell aus dem Landnutzungsparameterfile (z.B. landuse.par) bezieht. Im Einzelnen sind dies:

- 4 Stützwerte (d1 - d4) für den Leaf Area Index (*LAI*) für die Tage 110 (20. Apr.), 150 (30. Mai), 250 (7. Sep.) und 280 (7. Okt)
- 4 Stützwerte für effektive Wuchshöhe (*effH*) für die Tage 110 (20. Apr.), 150 (30. Mai), 250 (7. Sep.) und 280 (7. Okt)
- 12 Monatswerte für die Stomatawiderstände der entsprechenden Vegetation

Das Modell geht bei diesen Werten von einer Höhe von 400 m ü.N.N. aus. Entsprechend der tatsächlichen Höhe der jeweiligen Modelleinheit erfolgt eine Anpassung der Werte nach:

$$d'_x = dx + 0.025 \times (elevation - 400)$$

Hierdurch wird die Vegetationslänge bei tiefer liegenden Flächen verlängert, bei höher liegenden Flächen verkürzt. Zwischen den Stützstellen erfolgt eine diskrete Berechnung für jeden Modellzeitschritt durch lineare Interpolation.

Zu weiterführenden Informationen siehe auch: [ Krause 2001 ]

## CalcLanduseStateVars\_trop

### Metainformationen

Klassifikation	
----------------	--



Paket	org.unijena.j2k.inputData
Autor	Peter Krause, changes by Manfred Fink
Version	1.0_0
Datum	2011-05-30

## Attribute

### Eingabe

Name	Beschreibung	Einheit	Typ
entities	The current spatial entity		EntityCollection
elevation	elevation of the spatial modelling entity	m	Double
tempRes	temporal resolution [d   h   m]		String

### Ausgabe

Name	Beschreibung	Einheit	Typ
LAIArray	Array with LAI values for a standard year		DoubleArray
effHArray	Array with eff. Height values for a standard year		DoubleArray
rsc0Array	Monthly stomata resistance values	s / m	DoubleArray

## CalcLatLong

Eine Komponente zur Berechnung der extraterrestrischen Einstrahlung.

## Metainformationen

Klassifikation	
Paket	org.unijena.j2k.geographicalCalculations
Autor	Peter Krause
Version	1.0_0
Datum	2011-05-30

## Attribute

### Eingabe

Name	Beschreibung	Einheit	Typ
x	Die X Koordinate (Rechtswert) der räumlichen Modelleinheit.		Double
y	Die Y Koordinate (Hochwert) der räumlichen Modelleinheit.		Double

Name	Beschreibung	Einheit	Typ
slope	Die Neigung der räumlichen Modelleinheit.	degree	Double
aspect	Die Exposition der räumlichen Modelleinheit.	degree from north	Double
projection	Die Projektion der Koordinaten der räumlichen Modelleinheit.		String
tempRes	Die zeitliche Auflösung der Modellierung - (h) stündlich, (d) täglich, (m) monatlich.		String

## Ausgabe

Name	Beschreibung	Einheit	Typ
latitude	Die geographische Breite der räumlichen Modelleinheit	degree	Double
longitude	Die geographische Länge der räumlichen Modelleinheit	degree	Double
slAsCfArray	Korrekturfaktoren für die Neigungs-Expositions Kombination der räumlichen Modelleinheit.	n/a	DoubleArray

## Beschreibung der Komponente

In dieser Komponente werden projizierte Koordinaten in dezimale Gradangaben umgerechnet. Hierzu werden dem Modul die projizierten X und Y Koordinaten gemeinsam mit der Beschreibung der Projektion übergeben. Die Projektion wird in der Variablen "*projection*" abgelegt. Gültige Einträge sind hier:

- GK für Gauss-Krüger Projektion
- UTMxxZ für UTM Projektionen wobei xx für die Zonennummer und Z für den Zonenbuchstaben steht (z.B. UTM32U für Deutschland)
- LL für geographische Koordinaten in Dezimalgrad

Außerdem berechnet die Komponente einen zeitlich variablen Korrekturfaktor der zur Anpassung des Strahlungsempfang der räumlichen Modelleinheiten entsprechend ihrer Expositions- und Neigungskombination dient.

## CalcRainSnowParts

Eine Komponente zur Teilung des gefallenen Niederschlags in einen Regen- und einen Schneeanteil basierend auf der Lufttemperatur.

## Metainformationen

Klassifikation	
Paket	org.unijena.j2k.snow

Autor	Peter Krause
Version	1.0_0
Datum	2011-05-30

## Attribute

### Eingabe

Name	Beschreibung	Einheit	Typ
area	Die Fläche der räumlichen Modelleinheit [m²]	m²	Double
snow_trs	Temperatur bei der 50% des Niederschlags als Regen und 50% als Schnee fällt. [°C]	°C	Double
snow_trans	Temperaturintervall um snow_trs [K]	K	Double
tmin	Die minimale Lufttemperatur [°C]	°C	Double
tmean	Die mittlere Lufttemperatur [°C]	°C	Double
precip	Der Niederschlag [mm]	mm	Double

### Ausgabe

Name	Beschreibung	Einheit	Typ
rain	Die Regenmenge am Niederschlag [l]	L	Double
snow	Die Schneemenge am Niederschlag [l]	L	Double
svRain	Kopie von rain - nur zu Bilanzierungszwecken [l]	L	Double
svSnow	Kopie von snow - nur zu Bilanzierungszwecken [l]	L	Double

## Beschreibung der Komponente

Dieses Modul bestimmt anhand der Lufttemperatur, wie hoch der Schneeanteil am Gesamtniederschlag ist. Zur Bestimmung des Anteils wird angenommen, daß bei Unterschreiten einer bestimmten Grenztemperatur der gesamte Niederschlag als Schnee fällt und bei Überschreiten einer zweiten Grenztemperatur der gesamte Niederschlag als Regen fällt. Im Bereich zwischen diesen Grenztemperaturen treten Mischniederschläge auf. Zur Bestimmung der Grenztemperaturen und damit der Breite des Übergangsbereiches muß ein Temperaturwert ( $TRS$  in °C) angegeben werden, der der Temperatur entspricht, bei der 50% des Niederschlages als Schnee und 50% als Regen fallen. Zusätzlich muß ein Parameter  $Trans$  (in K) bestimmt werden, der der halben Breite des Übergangsbereiches entspricht (Abbildung 1). Der tatsächliche Schneeanteil ( $p(s)$ ) am Tagesniederschlag in Abhängigkeit von der Lufttemperatur ( $T$ ) berechnet sich dabei nach:

$$p(s) = \frac{TRS + Trans - T}{2 \times Trans} \text{ [mm]}$$

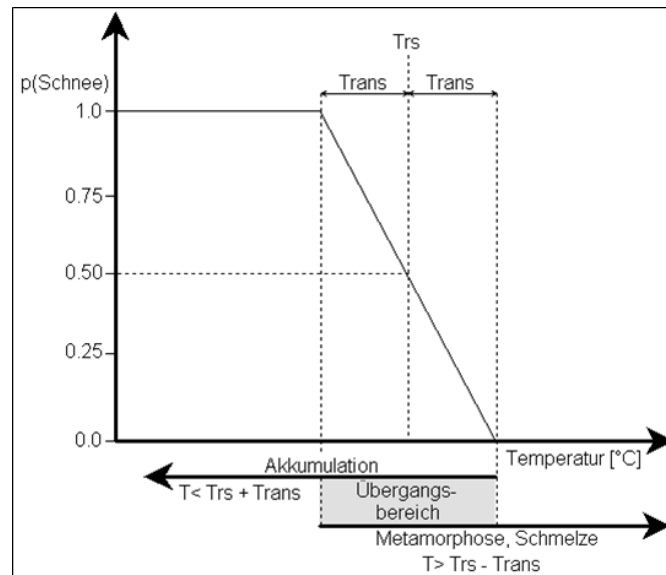
Die tägliche Schneemenge ( $N_s$ ) bzw. Regenmenge ( $N_r$ ) ergibt sich nach:

$$N_s = N \times p(s) \text{ [mm]}$$

$$N_r = N \times (1 - p(s)) \text{ [mm]}$$

Schließlich werden  $N_s$  und  $N_r$  noch mit der Fläche (area) der Modelleinheit multipliziert um mm in Liter umzuwandeln.

**Abbildung 1. Die Funktion zur Bestimmung des von Regen- bzw. Schneeanteils eines Niederschlagsereignisses**



## CalcRelativeHumidity

Eine Komponente zur Berechnung der relativen Feuchte aus der Lufttemperatur und der absoluten Feuchte.

## Metainformationen

Klassifikation	
Paket	org.unijena.j2k.inputData
Autor	Peter Krause
Version	1.0_0
Datum	2011-05-30

## Attribute

## Eingabe

Name	Beschreibung	Einheit	Typ
tmean	Die Lufttemperatur U[°C]	°C	Double
ahum	Die absolute Luftfeuchtigkeit U[g/m³]	g / m³	Double

## Ausgabe

Name	Beschreibung	Einheit	Typ
rhum	Die relative Luftfeuchtigkeit U[%]	%	Double

## Beschreibung der Komponente

Aus der absoluten Luftfeuchtigkeit (*ahum*) wird in Abhängigkeit von der Luftfeuchtigkeit die relative Luftfeuchtigkeit berechnet. Hierzu wird zunächst der Sättigungsdampfdruck (*es*) berechnet:

$$es = 6.108 \times e^{\frac{17.27 \times t_{mean}}{273.3 + t_{mean}}} \text{ [hPa]}$$

Daraus ergibt sich die maximal mögliche Luftfeuchtigkeit (*mhum*) nach:

$$mhum = es \times \frac{216.7}{t_{mean} + 273.15} \text{ [g/m}^3\text{]}$$

Und schließlich die relative Luftfeuchtigkeit (*rhum*) nach:

$$rhum = \frac{ahum}{mhum} \times 100 \text{ [%]}$$

## DoubleConditionalContext

### Metainformationen

Klassifikation	
Paket	jams.components.conditional
Autor	Sven Kralisch
Version	1.0_1
Datum	9. April 2008

## Attribute

### Eingabe

Name	Beschreibung	Einheit	Typ
value1	Double attribute "value1"		Double
value2	Double attribute "value2"		Double

## DoubleSetter

Eine Komponente zum Einfügen eines doppelten Attributs zu einem bestimmten Wert.

## Metainformationen

Klassifikation	
Paket	org.unijena.j2k
Autor	Sven Kralisch
Version	1.0_0
Datum	2012-09-26

## Attribute

### Eingabe

Name	Beschreibung	Einheit	Typ
value	doppelter Wert		Double[]

### Ausgabe

Name	Beschreibung	Einheit	Typ
attributes	die zu setzenden doppelten Attribute		Double[]

## Beschreibung der Komponente

Diese sehr einfache Komponente setzt die Werte der Attribute, die durch definiert werden, auf den Wert des Attributwerts. Die Komponente kann für das Zurücksetzen räumlicher/zeitlicher Ausgleichswerte genutzt werden nach einem Schritt der räumlichen/zeitlichen Iteration.

## DoubleTransfer

### Metainformationen

Klassifikation	
Paket	jams.components.datatransfer
Autor	Sven Kralisch
Version	1.0_0
Datum	2012-09-27

## Attribute

### Eingabe

Name	Beschreibung	Einheit	Typ
target	Target entity		Entity

Name	Beschreibung	Einheit	Typ
inNames	Target entity's receiving attributes		String[]

## Status

Name	Beschreibung	Einheit	Typ
values	Data to be transferred		Double[]

# GeomReader

## Metainformationen

Klassifikation	
Paket	jams.components.io
Autor	Christian Schwartze
Version	1.0_0
Datum	2010-10-22

## Attribute

### Eingabe

Name	Beschreibung	Einheit	Typ
shapeFileName	Dateipfad zum HRU-Shapefile		String
hrus	HRU-Objekte		EntityCollection
idName	Name der ID-Spalte im HRU-Shapefile		String

### Ausgabe

Name	Beschreibung	Einheit	Typ
baseShape	null		String

## Beschreibung der Komponente

Die Komponente dient zum Einlesen der Geometrien eines HRU-Shapefiles, um sie im Anschluss mit den JAMS Entities zu verknüpfen. Auf diese Weise erhalten die HRU-Objekte (JAMSEntityCollection) ihr Geometrie-Attribut.

# IDWWeightCalculator

Eine Komponente zur Berechnung von Gewichtswerten für das Inverse-Distanzen-Verfahren.

## Metainformationen

Klassifikation	
Paket	org.unijena.j2k.regionalisation
Autor	Peter Krause and Sven Kralisch
Version	1.0_0
Datum	2008-12-07

## Attribute

### Eingabe

Name	Beschreibung	Einheit	Typ
entityX	Einheit x-Koordinate		Double
entityY	Einheit y-Koordinate		Double
pidw	Stärke der IDW-Funktion		Double
equalWeights	Gewichte für Thiessen-Polygone		Boolean
latLong	Berechnung mit geografischen Koordinaten (lat, long)		Boolean
dataStoreID	ID des Datenspeichers zum Einlesen von Stationskoordinaten		String

### Ausgabe

Name	Beschreibung	Einheit	Typ
statWeights	Gewichte des IDW-Anteils der Regionalisierung		DoubleArray
statOrder	Positionsmatrix zur Bestimmung der besten Gewichte		IntegerArray

## Beschreibung der Komponente

Diese Komponente berechnet die gewichteten Werte für die Regionalisierung auf Basis des *Inverse Distance Weighting (IDW)* Verfahren. Für diesen Zweck werden die Koordinaten (`entityX/entityY`) einer Modelleinheit und die Anzahl der Stationskoordinaten von einem Datenspeicher auf Grund der ID (`dataStoreID`) identifiziert. Diese werden als Input für die Berechnung der Distanzen zu diesen Stationen benutzt.

Als ein Ergebnis berechnet die Komponente:

1. einen gewichteten Wert für jede Station (`statWeights`), mit höheren Werten bei räumliche näheren Stationen,
2. eine Matrix von Stationskoordinaten (`statOrder`) welche nach der Distanz zur Einheit sortiert sind.

Um die Gewichtung zu berechnen müssen zusätzliche Parameter bereitgestellt werden: ein Gewichtungswert (`pidw`), zur Betonung oder Abschwächung der Distanz zwischen den Stationen, ein



Kennzeichen (*latLong*) ob ein geographisches Koordinatensystem benötigt wird und ein Kennzeichen (*equalWeights*) welches die Berechnung der gleichen Gewichtung verstärkt (für *n* Stationen hat jede Station die Gewichtung  $1/n$ ).

Diese Komponente wird hauptsächlich für die Vorverarbeitung genutzt, wenn die IDW Gewichtung nur für jede räumliche Einheit berechnet werden muss.

## InitJ2KProcessGroundwater

Eine Komponente zur Initialisierung der beiden Grundwasserspeicher des Grundwassermoduls.

### Metainformationen

Klassifikation	
Paket	org.unijena.j2k.groundwater
Autor	Peter Krause
Version	1.0_0
Datum	2010-10-29

### Attribute

#### Eingabe

Name	Beschreibung	Einheit	Typ
entities	Der Satz der Modelleinheiten.		EntityCollection
initRG1	null	n/a	Double
initRG2	Die relative Speicherfüllung der Grundwasserkomponente RG2 beim Modellstart.	n/a	Double

#### Ausgabe

Name	Beschreibung	Einheit	Typ
maxRG1	Die maximale Speicherkapazität für die Grundwasserkomponente RG1.	L	Double
maxRG2	Die maximale Speicherkapazität für die Grundwasserkomponente RG2.	L	Double
actRG1	Die aktuelle Speicherkapazität für die Grundwasserkomponente RG1.	L	Double
actRG2	Die aktuelle Speicherkapazität für die Grundwasserkomponente RG2.	L	Double

## Beschreibung der Komponente

Diese Komponente initialisiert die beiden Grundwasserspeicher *RG1* und *RG2* des Grundwassermoduls "J2KProcessGroundwater". Hierzu wird die maximale Speicherkapazität der beiden Speicher, die aus

dem entsprechenden Parameterfile (z.B. "hgeo.par") stammen und dort in der Einheit mm abgelegt sind, mit der Fläche der jeweiligen Modelleinheit multipliziert um eine absolute Speicherkapazität in Litern zu erhalten. Diese Werte werden in den Attributen *maxRG1* und *maxRG2* abgelegt. Zusätzlich kann die aktuelle Speicherfüllung der beiden Grundwasserkomponenten (*actRG1* und *actRG2*) noch mit einer Anfangsfüllung versehen werden. Dies erfolgt über die beiden Parameter *initRG1* und *initRG2*, die eine relative Füllung im Verhältnis zur maximalen Kapazität beschreiben.

# InitJ2KProcessLumpedSoilWaterStates

Eine Komponente zur Initialisierung des ungelayernten Bodenmoduls.

## Metainformationen

Klassifikation	
Paket	org.unijena.j2k.soilWater
Autor	Peter Krause
Version	1.0_0
Datum	2010-10-29

## Attribute

### Eingabe

Name	Beschreibung	Einheit	Typ
entities	der Satz der Modelleinheiten		EntityCollection
area	die Fläche der aktuellen Modelleinheit	m <sup>2</sup>	Double
FCAdaptation	Eichparameter mit dem die Speicherkapazität des Mittel- und Feinporenspeichers (MPS) erhöht oder erniedrigt werden kann	n/a	Double
ACAdaptation	Eichparameter mit dem die Speicherkapazität des Grobporenspeichers (LPS) erhöht oder erniedrigt werden kann	n/a	Double
rootDepth	die effektive Wurzeltiefe der Landbedeckungseinheit der aktuellen Modelleinheit	dm	Double
satStartLPS	die relative Speichersättigung für den Grobporenspeicher (LPS) beim Modellstart	n/a	Double
satStartMPS	der relative Speichersättigung für den Mittel- und Feinporenspeicher (MPS) beim Modellstart	n/a	Double

### Ausgabe

Name	Beschreibung	Einheit	Typ
maxMPS	die maximale Speicherkapazität für den Mittel- und Feinporenspeicher (MPS)	L	Double

Name	Beschreibung	Einheit	Typ
maxLPS	die maximale Speicherkapazität für den Grobporenspeicher (LPS)	L	Double
actMPS	die aktuelle Speicherkapazität für den Mittel- und Feinporenspeicher (MPS)	L	Double
actLPS	die aktuelle Speicherkapazität für den Grobporenspeicher (LPS)	L	Double
satMPS	die relative Speichersättigung für den Mittel- und Feinporenspeicher (MPS)	n/a	Double
satLPS	die relative Speichersättigung für den Grobporenspeicher (LPS)	n/a	Double
satSoil	die relative Speichersättigung des Gesamtbodens (LPS + MPS)	n/a	Double

## Beschreibung der Komponente

Diese Komponente initialisiert das ungelayerete Bodenmodul (J2KProcessLumpedSoilWater). Hierzu weist es jeder Modelleinheit spezifische Bodenfeuchtespeicher zu (*actMPS*, *actLPS*), parametrisiert diese in Abhängigkeit des vorherrschenden Bodentyps (*maxMPS*, *maxLPS*) (z.B. aus *soil.par*) sowie der Landnutzung. Hierzu wird für jeden Dezimeter innerhalb der effektiven Durchwurzelungstiefe die Feldkapazität aufsummiert und die kumulierte Summe als maximale Speicherkapazität des Mittelporenspeichers (*maxMPS*) der Modelleinheit zugewiesen. Die maximale Speicherkapazität des Grobporenspeichers (*maxLPS*) wird für den gesamten Boden berechnet. Zusätzlich können den Speichern relative Anfangswassergehalte zugewiesen werden. Schließlich werden die Wassergehalte von mm in Litern umgerechnet indem mit der Fläche (in m<sup>2</sup>) der jeweiligen Modelleinheit multipliziert wird.

Zu weiterführenden Informationen siehe auch: *Krause 2001*

## J2KArrayGrabber

### Metainformationen

Klassifikation	
Paket	org.unijena.j2k
Autor	Peter Krause
Version	1.0_0
Datum	2011-05-30

### Attribute

### Eingabe

Name	Beschreibung	Einheit	Typ
tempRes	temporal resolution [m   d   h]		String
time	time		Calendar

Name	Beschreibung	Einheit	Typ
extRadArray	extraTerrRadiationArray		DoubleArray
LAIArray	LeafAreaIndexArray		DoubleArray
effHArray	EffectiveHeightArray		DoubleArray
rsc0Array	rsc0 Array		DoubleArray
slAsCfArray	slopeAspectCorrectionFactorArray		DoubleArray
haudeFactorArray	Haude factor array		DoubleArray

## Ausgabe

Name	Beschreibung	Einheit	Typ
actExtRad	actExtraTerrRadiation		Double
actLAI	actLAI		Double
actEffH	actEffH		Double
actRsc0	actRsc0		Double
actHaudeFactor	actHaudeFactor		Double
actSlAsCf	actSlopeAspectCorrectionFactor		Double

# J2K Multiplication

## Metainformationen

Klassifikation	
Paket	org.unijena.j2k.mathematicalCalculations
Autor	Manfred Fink
Version	1.0_0
Datum	2012-03-19

## Attribute

## Eingabe

Name	Beschreibung	Einheit	Typ
mult1	multiplier 1	-	Double
mult2	multiplier 2	-	Double

## Ausgabe

Name	Beschreibung	Einheit	Typ
Product	Product	-	Double

# J2KProcessGroundwater

Eine Komponente zur Beschreibung der Grundwasserdynamik.

## Metainformationen

Klassifikation	
Paket	org.unijena.j2k.groundwater
Autor	Peter Krause
Version	1.0_1
Datum	2013-04-26

## Attribute

### Eingabe

Name	Beschreibung	Einheit	Typ
slope	Die Hangneigung der Modelleinheit [Grad]	deg	Double
maxRG1	Die maximale Speichergröße für den schnellen Grundwasserspeicher RG1 [l]	L	Double
maxRG2	Die maximale Speichergröße für den langsamen Grundwasserspeicher RG2 [l]	L	Double
kRG1	Rückhaltekoefizient k für die schnelle Grundwasserkomponente RG1 [1/ts]		Double
kRG2	Rückhaltekoefizient k für die langsame Grundwasserkomponente RG2 [1/ts]		Double
percolation	Die Perkolation aus dem ungesättigten Bodenbereich [l]	L	Double
maxSoilStorage	Die maximale Wasserspeicherkapazität des ungesättigten Bodenbereichs [l]	L	Double
actSoilStorage	Die tatsächliche Wasserspeicherkapazität des ungesättigten Bodenbereichs [l]	L	Double
gwRG1Fact	Kalibrierparameter für den Abfluss aus dem schnellen Grundwasserspeicher RG1 WB[0 .. inf] [-]		Double
gwRG2Fact	Kalibrierparameter für den Abfluss aus dem langsamen Grundwasserspeicher RG2 WB[0 .. inf] [-]		Double
gwRG1RG2dist	Kalibrierparameter zur Verteilung der Perkulationsrate auf die beiden Grundwasserspeicher RG1 und RG2 WB [0 .. inf] [-]		Double
gwCapRise	Kalibrierparameter zur Berechnung des kapillaren Aufstiegs WB[0-inf] [-]		Double

## Status

Name	Beschreibung	Einheit	Typ
actRG1	Die tatsächliche Speicherfüllung für den schnellen Grundwasserspeicher RG1 [l]	L	Double
actRG2	Die tatsächliche Speicherfüllung für den langsamen Grundwasserspeicher RG2 [l]	L	Double
inRG1	Der laterale Zufluss in den schnellen Grundwasserspeicher RG1 aus benachbarten Flächen [l]	L	Double
inRG2	Der laterale Zufluss in den langsamen Grundwasserspeicher RG2 aus benachbarten Flächen [l]	L	Double
gwExcess	Wassermenge, die aufgrund begrenzter Speicher nicht im Grundwasserbereich verbleiben kann [l]	L	Double

## Ausgabe

Name	Beschreibung	Einheit	Typ
outRG1	Abfluss aus dem schnellen Grundwasserspeicher RG1 [l]	L	Double
outRG2	Abfluss aus dem langsamen Grundwasserspeicher RG2 [l]	L	Double
genRG1	wie RG1out [l]	L	Double
genRG2	wie RG2out [l]	L	Double

## Beschreibung der Komponente

Das Modellkonzept des Grundwassermoduls in J2000 ermöglicht, unter Berücksichtigung der unterschiedlichen Speicher- und Abflussverhalten, die Betrachtung des Grundwasserabflusses aller im Einzugsgebiet vorkommenden geologischen Formationen, die einzeln parametrisiert werden müssen. In den einzelnen geologischen Einheiten wird zwischen dem oberen Grundwasserspeicher (*RG1*) im lockeren Verwitterungsmaterial mit hohen Durchlässigkeiten und kurzen Verweilzeiten und dem unteren Grundwasserspeicher (*RG2*) in der Matrix und in Rissen und Klüften des Grundgesteins mit niedrigeren Durchlässigkeiten und höheren Verweilzeiten unterschieden. Es werden dementsprechend zwei Basisabflusskomponenten generiert, eine schnelle aus dem oberen Grundwasserspeicher und eine langsame aus dem unteren Grundwasserspeicher. Die Füllung der Grundwasserspeicher erfolgt aus der vertikalen Abflusskomponente des Bodenmoduls (Perkolations), die Entleerung kann durch die lateralen unterirdischen Abflusskomponenten und kapillaren Aufstieg in die ungesättigte Zone erfolgen. Die Parametrisierung der Grundwasserspeicher erfolgt mit der Bestimmung der maximalen Speicherkapazität des oberen (*maxRG1*) und des unteren Grundwasserspeichers (*maxRG2*) sowie jeweils eines Rückhaltekoefizienten für die beiden Speicher, (*kRG1*) und (*kRG2*). Die maximale Speicherkapazität lässt sich aus dem Produkt des Hohlraumanteils und der Mächtigkeit des einzelnen Speichers pro m<sup>2</sup> Einheitsfläche abschätzen. Beide Parameter sind für jede geologische Einheit separat zu bestimmen und in einem entsprechenden Parameterfile abzulegen. Folgende Abbildung (Abbildung 2) zeigt ein Beispiel für solch ein Parameterfile:

## Abbildung 2. hgeo.par

#hgeo.par	GID	RG1_max	RG2_max	RG1_k	RG2_k	RG1_active
	1	0	0	0	0	0
	-9999	-9999	-9999	-9999	-9999	1
	n/a	mm	mm	d	d	n/a
	6	90	220	8	50	1
	8	90	1050	12	572	1
	15	80	825	14	677	1
	16	70	750	16	671	1
	18	70	750	16	671	1
	5	70	750	16	671	1
	7	70	750	16	671	1
	17	70	750	16	671	1
	9	70	750	16	671	1
	19	70	750	16	671	1
#end of hydrogeo.par						

Der erste Prozess der berechnet wird ist der kapillare Aufstieg. Hierzu wird angenommen, dass dieser aus dem langsamen Grundwasserspeicher *RG2* erfolgt. Die Berechnung erfolgt anhand der tatsächlichen Füllung dieses Speichers *actRG2*, dem Feuchtedefizit (*deltaSoilStor*) bzw. der relativen Sättigung (*satSoilStor*) im oberhalb liegenden Boden und dem Eichparameter *gwCapRise*. Kapillarer Aufstieg wird dabei nur berechnet wenn die aktuelle Speicherfüllung des Grundwasserspeichers *RG2* das Bodenfeuchtedefizit übersteigt und der Eichparameter *gwCapRise* einen Wert größer Null besitzt. Durch Nullsetzen lässt sich der kapillare Aufstieg unterbinden.

Wird kapillarer Aufstieg berechnet wird hierzu zunächst die Aufstiegsrate (*insoilStor*) berechnet:

$$inSoilStor = deltaSoilStor \times \left(1 - e^{-\frac{1 \times gwCapRise}{satSoilStor}}\right)$$

Diese wird dann dem Grundwasserspeicher *RG2* entnommen und dem oberhalb liegenden Bodenwasserspeicher hinzuaddiert.

Nach der Berechnung des kapillaren Aufstiegs erfolgt die Berücksichtigung des lateralen Zustroms. Hierzu werden die beiden Zuflusskomponenten *inRG1* und *inRG2* den jeweiligen Speichern *actRG1* und *actRG2* hinzugefügt. Wird hierdurch die maximale Speichergröße dieser beiden Speicher überschritten, wird überschüssiges Wasser direkt an die Ausflusskomponente weitergegeben.

Die Aufteilung der Perkolation aus dem oben liegenden Boden erfolgt in Abhängigkeit von der Hangneigung (*slope*) der jeweiligen Modelleinheit und einem Eichkoeffizienten (*gwRG1RG2dist*) nach folgenden Einzelschritten. Zunächst wird der Einfluss der Hangneigung berücksichtigt:

$$slope\_weight = \tan\left(slope \times \frac{\pi}{180}\right) [-]$$

$$gradh = (1 - slope\_weight) \times gwRG1RG2dist [-]$$

$$gradh = \min_1 gradh$$

Dann wird hiermit die Perkulationsrate aufgeteilt:

$$potRG1 = (1 - gradh) \times percolation$$

$$potRG2 = gradh \times percolation$$

Diese potentiellen Zuflüsse werden dann noch gegen die jeweiligen aktuellen Speicher abgeglichen. Kann *RG2* den Zufluss *pot\_RG2* nicht vollständig aufnehmen wird der Überschuss zu *pot\_RG1* weitergegeben.

Ist *RG1* nicht in der Lage *pot\_RG1* aufzunehmen wird der Überschuss der Variablen *gwExcess* zugewiesen. Je nach Modellkonzept kann diese Variable dann einer bestimmten Abflusskomponente zugewiesen werden.

Die Berechnung der Wasserabgabe erfolgt in Abhängigkeit der aktuellen Speicherfüllungen in Form einer linearen Auslauffunktion. Die Speicherrückhaltekoeffizienten (*kRG1*, *kRG2*), welche als Verweilzeiten des Wassers im betrachteten Speicher zu verstehen sind, gehen als Faktor des aktuellen Speicherinhaltes (*actRG1* und *actRG2*) in die Berechnung des Grundwasserausflusses (*outRG1* und *outRG2*) wie folgt ein:

$$outRG1 = \frac{1}{gwRG1Fact \times kRG1} \times actRG1 \text{ [mm]}$$

$$outRG2 = \frac{1}{gwRG2Fact \times kRG2} \times actRG2 \text{ [mm]}$$

Über die beiden Parameter *gwRG1Fact* und *gwRG2Fact* kann die Grundwasserdynamik kalibriert werden.

Zu weiterführenden Informationen siehe auch *Krause 2001*

## J2KProcessInterception

Eine Komponente zur Berechnung von Niederschlagsinterzeption an Pflanzenoberflächen.

### Metainformationen

Klassifikation	
Paket	org.unijena.j2k.interception
Autor	Peter Krause
Version	1.1_0
Datum	2012-10-10

### Attribute

### Eingabe

Name	Beschreibung	Einheit	Typ
area	Fläche der Modelleinheit	m <sup>2</sup>	Double
tmean	Mittlere Lufttemperatur	degC	Double
rain	Flüssiger Niederschlag	L	Double
snow	Fester Niederschlag	L	Double
potET	Potentielle Verdunstung	L	Double
actLAI	Blattflächenindex (aktuell)		Double
snow_trs	Temperaturparameter	degC	Double
snow_trans	Temperaturparameter	K	Double
a_rain	Eichparameter für Interzeption von flüssigem Niederschlag	mm	Double



Name	Beschreibung	Einheit	Typ
a_snow	Eichparameter für Interzeption von festem Niederschlag	mm	Double

## Status

Name	Beschreibung	Einheit	Typ
actET	Aktuelle Verdunstung	L	Double
intercStorage	Interzeptionsspeicherfüllung	L	Double

## Ausgabe

Name	Beschreibung	Einheit	Typ
netRain	Bestandsniederschlag (flüssig)	L	Double
netSnow	Bestandsniederschlag (fest)	L	Double
throughfall	Bestandsniederschlag (gesamt)	L	Double
interception	Interzeption	L	Double

## Beschreibung der Komponente

Das Interzeptionsmodul dient der Berechnung der Bestandesniederschläge aus den Freilandniederschlägen in Abhängigkeit von der jeweiligen Vegetationsbedeckung und ihrer unterschiedlichen Ausprägung im Jahresgang. Durch die Interzeption wird der Wasserhaushalt, je nach Art des Niederschlages, unterschiedlich beeinflusst. Zum einen ist dies die Speicherung von Niederschlagswasser an den oberirdischen Sprossteilen der Vegetation, das dann der Verdunstung zugeführt wird und für die nachfolgenden Prozesse nicht mehr zur Verfügung steht [Baumgartner1995]. Zum anderen kann durch die Interzeption eine beträchtliche Menge von Niederschlag in Form von Schnee durch die Vegetation kurz- bis mittelfristig zurückgehalten werden und erst verzögert, nach dem Abtauen, zum Boden gelangen und dort für die sich anschließenden Prozesse zur Verfügung stehen. Die maximale Interzeptionsspeicherkapazität ( $Int_{max}$ ) ist einerseits durch die Vegetationsart, insbesondere durch den Blattflächenindex im Jahresgang, andererseits von der Art der Niederschläge (ob Regen oder Schnee) abhängig und wird (nach [Dickinson1984]) folgendermaßen berechnet:

$$Int_{max} = \alpha \cdot LAI \quad [\text{mm}]$$

mit:

$\alpha$ : Speicherkapazität pro m<sup>2</sup> Blattfläche [mm]

$LAI$ : Blattflächenindex der betreffenden Landnutzungsklasse [-]

Der Parameter  $\alpha$  besitzt je nach Art (Schnee oder Regen) des interzeptierten Niederschlages unterschiedliche Ausprägungen. Dies ist notwendig, da die Interzeptionskapazität für Schnee deutlich über der von flüssigem Niederschlag liegt. Welche Ausprägung des Parameters für die tägliche Berechnung der maximalen Interzeptionsspeicherkapazität herangezogen wird, ist von der mittleren Lufttemperatur abhängig. Unterschreitet sie einen kritischen Wert (der vom Anwender bei den Parametern für die Schneedeckenberechnung anzugeben ist und bereits bei der Regionalisierung diskutiert wurde), wird der Niederschlag als Schneeniederschlag betrachtet. Liegt sie dagegen über dem kritischen Wert wird von Niederschlag in Form von Regen ausgegangen.

Durch die Interzeption wird der Freilandniederschlag, der auf die betreffende Fläche fällt, um den Interzeptionsteil auf den Bestandesniederschlag reduziert. Bestandesniederschlag tritt also nur dann auf, wenn die maximale Interzeptionsspeicherkapazität der Vegetation erschöpft ist. Der Überschuss wird dann als durchfallender Niederschlag an das folgende Modul weitergegeben.

Die Entleerung des Interzeptionsspeicher erfolgt normalerweise ausschließlich durch die Verdunstung. Ein Sonderfall tritt auf, wenn sich die Ausprägung des Parameter  $\alpha$  auf Grund der Lufttemperatur von Schnee auf Regen ändert. Dies führt zu einer sprunghaften Herabsetzung der maximalen Interzeptionsspeicherkapazität. Damit wird das Abtauen von auf der Vegetationsoberfläche gespeichertem Schnee nachgebildet. Eventueller Überschuss wird als abtropfender Niederschlag an das sich anschließende Modul weitergegeben.

## J2KProcessLumpedSoilWater\_DPSmod

### Metainformationen

Klassifikation	
Paket	soilWater
Autor	Peter Krause, DPS modified by Markus Meinhardt
Version	1.0_0
Datum	2011-05-30

### Attribute

### Eingabe

Name	Beschreibung	Einheit	Typ
time	time		Calendar
entity	The current spatial modelling entity		Entity
area	attribute area	m <sup>2</sup>	Double
slope	attribute slope	deg	Double
sealedGrade	sealed grade		Double
potET	state variable potET	L	Double
snowDepth	snow depth	mm	Double
snowMelt	daily snow melt	L	Double
maxMPS	maximum MPS	L	Double
maxLPS	maximum LPS	L	Double
soilMaxDPS	maximum depression storage	L	Double
soilPolRed	poly reduction of ETP		Double
soilLinRed	linear reduction of ETP		Double
soilMaxInfSummer	maximum infiltration rate in summer for one time step	mm	Double

Name	Beschreibung	Einheit	Typ
soilMaxInfWinter	maximum infiltration rate in winter for one time step	mm	Double
soilMaxInfSnow	maximum infiltration rate on snow for one time step	mm	Double
soilImpGT80	maximum infiltration part on sealed areas (gt 80%)		Double
soilImpLT80	maximum infiltration part on sealed areas (lt 80%)		Double
soilDistMPSLPS	MPS/LPS distribution coefficient for inflow		Double
soilDiffMPSLPS	MPS/LPS diffusion coefficient		Double
soilOutLPS	LPS outflow coefficient		Double
soilLatVertLPS	LPS lateral-vertical distribution coefficient		Double
soilMaxPerc	maximum percolation rate [mm/d]	mm d <sup>-1</sup>	Double
soilConcRD1	concentration coefficient for RD1		Double
soilConcRD2	concentration coefficient for RD2		Double

## Status

Name	Beschreibung	Einheit	Typ
netRain	state variable net rain	L	Double
netSnow	state variable net snow	L	Double
actET	state variable actET	L	Double
actMPS	state var actual MPS	L	Double
actLPS	state var actual LPS	L	Double
actDPS	state var actual depression storage	L	Double
satMPS	state var saturation of MPS		Double
satLPS	state var saturation of LPS		Double
satSoil	state var saturation of whole soil		Double
inRD1	statevar RD1 inflow	L	Double
inRD2	statevar RD2 inflow	L	Double

## Ausgabe

Name	Beschreibung	Einheit	Typ
infiltration	statevar infiltration	L	Double
interflow	statevar interflow	L	Double
percolation	statevar percolation	L	Double
outRD1	statevar RD1 outflow	L	Double
genRD1	statevar RD1 generation	L	Double
outRD2	statevar RD2 outflow	L	Double

Name	Beschreibung	Einheit	Typ
genRD2	statevar RD2 generation	L	Double

## J2KProcessReachRouting

Eine Komponente zur Modellierung der Fließvorgänge im Vorfluternetzwerk des Einzugsgebietes.

### Metainformationen

Klassifikation	
Paket	org.unijena.j2k.routing
Autor	Peter Krause
Version	1.0_1
Datum	2011-05-30

### Attribute

#### Eingabe

Name	Beschreibung	Einheit	Typ
entities	Die Kollektion der Flussabschnitt Elemente (river reaches)		EntityCollection
length	Die Länge eines Flussabschnitts [m]	m	Double
slope	Das Gefälle eines Flussabschnitts [(Starthöhe-Endhöhe)/Länge]	%	Double
slopeAsProportion	null		Boolean
width	Die Breite eines Flussabschnitts [m]	m	Double
roughness	Die Sohlrauhigkeit nach Maning-Strickler [-]		Double
flowRouteTA	Eichparameter zur Anpassung der Fließgeschwindigkeit [0 .. posInf]		Double
tempRes	Die zeitliche Modellauflösung [d oder h]		String

### Status

Name	Beschreibung	Einheit	Typ
inRD1	Zufluss der Komponente RD1 in den Flussabschnitt	L	Double
inRD2	Zufluss der Komponente RD2 in den Flussabschnitt	L	Double
inRG1	Zufluss der Komponente RG1 in den Flussabschnitt	L	Double

Name	Beschreibung	Einheit	Typ
inRG2	Zufluss der Komponente RG2 in den Flussabschnitt	L	Double
inAddIn	Zusätzlicher Zufluss z.B. aus oberliegenden Einzugsgebieten	L	Double
actRD1	Die aktuelle Speicherung der Komponente RD1 im Flussabschnitt	L	Double
actRD2	Die aktuelle Speicherung der Komponente RD2 im Flussabschnitt	L	Double
actRG1	Die aktuelle Speicherung der Komponente RG1 im Flussabschnitt	L	Double
actRG2	Die aktuelle Speicherung der Komponente RG2 im Flussabschnitt	L	Double
actAddIn	Die aktuelle Speicherung von zusätzlichem Zufluss im Flussabschnitt	L	Double
channelStorage	Die Summe der aktuellen Speicherungen	L	Double

## Ausgabe

Name	Beschreibung	Einheit	Typ
outRD1	Abfluss der Komponente RD1 aus dem Flussabschnitt	L	Double
outRD2	Abfluss der Komponente RD2 aus dem Flussabschnitt	L	Double
outRG1	Abfluss der Komponente RG1 aus dem Flussabschnitt	L	Double
outRG2	Abfluss der Komponente RG2 aus dem Flussabschnitt	L	Double
outAddIn	Abfluss zusätzlichen Zuflusses aus dem Flussabschnitt	L	Double
simRunoff	Die Summe der Abflüsse aus dem Flussabschnitt	L	Double
catchmentRD1	Speicherung der Komponente RD1 (nur Gebietsauslass)	L	Double
catchmentRD2	Speicherung der Komponente RD2 (nur Gebietsauslass)	L	Double
catchmentRG1	Speicherung der Komponente RG1 (nur Gebietsauslass)	L	Double
catchmentRG2	Speicherung der Komponente RG2 (nur Gebietsauslass)	L	Double
catchmentAddIn	Speicherung der Zusatzkomponente (nur Gebietsauslass)	L	Double
catchmentSimRunoff	Summe der Speicherungen (nur Gebietsauslass)	L	Double
waterLevel	null		Double

## Beschreibung der Komponente

Diese Komponente dient der Modellierung der Fließvorgänge im Vorfluternetzwerk des Einzugsgebietes. Hierzu müssen zunächst einzelne Flussabschnitte parametrisiert werden. Die einzelnen Parameter, die zugewiesen werden müssen sind die Länge in Metern, die Neigung in %, die durchschnittliche Breite in Metern und die Sohlrauhigkeit nach Manning-Strickler. Diese werden in einer Parameterdatei abgelegt, die vom Modell eingelesen und daraus die entsprechenden Objekte erzeugt werden. Zusätzlich enthält diese Datei noch Informationen zum Aufbau der Fließtopologie, indem für jeden Flussabschnitt die ID vermerkt wird, in der er wiederum mündet. Der Flussabschnitt, der den Gebietsauslass repräsentiert, muss die ID 0 erhalten.

Die einzelnen Flussabschnitte erhalten ihr Wasser durch angrenzende räumliche Modelleinheiten und oberliegende Flussabschnitte. Im Flussabschnitt wird für das Wasservolumen eine Fließgeschwindigkeit berechnet und dann auf Basis dieser Fließgeschwindigkeit und der Vorfluterlänge ein entsprechender Teil des Volumens an den nächsten unterliegenden Flussabschnitt abgegeben. Für die Berechnung wird vereinfacht ein Rechteckquerschnitt angenommen. Obwohl das komplette Wasservolumen geroutet wird, werden dennoch die relativen Anteile der einzelnen Abflusskomponenten erhalten, so dass diese jederzeit und in jedem Flussabschnitt gesondert betrachtet werden können.

Das Modul beschreibt die Fließvorgänge im Gerinne mittels eines kinematischen Wellenansatz und der Berechnung der Fließgeschwindigkeit nach Manning & Strickler. Der einzige einzustellende Parameter ( $TA$ ) ist ein vom Anwender zu bestimmender Routingkoeffizient. Er repräsentiert die Laufzeit der Abflusswelle, welche sich nach einem Niederschlagsereignis im Gerinne bis zum Gebietsauslass bewegt. Sein Wert geht neben der Fließgeschwindigkeit des Gewässers ( $v$ ) und der Fließlänge ( $fl$ ) in die Berechnung eines Abflussrückhaltekoefizienten ( $Rk$ ) ein:

$$Rk = \frac{v}{fl} \times TA \times 3600 [-]$$

Zuvor ist jedoch die Fließgeschwindigkeit ( $v_{new}$ ) mit dem Rauigkeitsfaktor von Manning ( $M$ ), dem Gefälle des Flussbettes ( $l$ ) und dem hydraulischen Radius ( $Rh$ ) zu bestimmen. Der hydraulische Radius wird wiederum aus dem durchflossenen Querschnitt ( $A$ ) des Flussabschnittes, berechnet, der sich aus dem Durchfluss ( $q$ ), der Fließgeschwindigkeit ( $v$ ) und der Flussbettbreite ( $b$ ) ergibt. Bei diesem Ansatz wird zunächst eine Ausgangsgeschwindigkeit ( $v_{init}$ ) von 1 m/s angenommen, welche dann iterativ mit der neu berechneten Fließgeschwindigkeit ( $v_{new}$ ) abgeglichen wird, bis die Abweichung der beiden Geschwindigkeiten einen Wert kleiner als 0,001 m/s beträgt:

$$Rh = \frac{A}{b + 2 \frac{A}{b}} [m]$$

mit:

$$A = \frac{q}{v_{init}} [m^2]$$

$$v_{new} = M \times Rh^{\frac{2}{3}} \times l^{\frac{1}{3}} [m^3/s]$$

Schließlich wird mit dem ermittelten Ausflussrückhaltekoefizienten ( $Rk$ ) der Anteil des Wasservolumens des jeweiligen Flussabschnittes ( $q_{act}$ ), der zum Abfluss kommt ( $q$ ), berechnet.

$$q = q_{act} \times e^{\frac{-1}{Rk}} [m^3/s]$$

Je höher der angenommene Wert von  $TA$  ist, desto schneller bewegt sich die Abflusswelle innerhalb eines bestimmten Zeitabschnittes und umso weniger Wasser verbleibt im Gerinne.

Weiterführende Informationen in *Krause et al. 2005* .

# J2KProcessSnow

Eine Komponente zur Berechnung der Schneeschmelze.

## Metainformationen

Klassifikation	
Paket	org.unijena.j2k.snow
Autor	Peter Krause
Version	1.0_0
Datum	2011-05-30

## Attribute

### Eingabe

Name	Beschreibung	Einheit	Typ
time	null		Calendar
entity	null		Entity
area	Die Fläche der Modelleinheit	m <sup>2</sup>	Double
actSIAsCf	Der Neigungs-Expositions-Korrekturfaktor		Double
minTemp	Die minimale Lufttemperatur (optional)	°C	Double
meanTemp	Die mittlere Lufttemperatur (optional)	°C	Double
maxTemp	Die maximale Lufttemperatur (optional)	°C	Double
baseTemp	Grenztemperatur ab der Metamorphose und Schmelze stattfindet	°C	Double
t_factor	Schmelzfaktor der den Energieeintrag über fühlbare Wärme beschreibt	mm °C <sup>-1</sup>	Double
r_factor	Schmelzfaktor der den Energieeintrag über flüssigen Niederschlag beschreibt	°C <sup>-1</sup>	Double
g_factor	Schmelzfaktor der den Energieeintrag über den Bodenwärmestrom beschreibt	mm	Double
snowCritDens	Kritische Schneedichte ab der Wasser nicht länger in der Schneedecke zurückgehalten werden kann	g cm <sup>-3</sup>	Double
ccf_factor	Faktor zur Berechnung des Kältegehaltes [mm/°C]		Double
active	null		Boolean

## Status

Name	Beschreibung	Einheit	Typ
netRain	Der flüssige Anteil des Niederschlags	L	Double

Name	Beschreibung	Einheit	Typ
netSnow	Der feste Anteil des Niederschlags	L	Double
snowTotSWE	Das Gesamtschneewasseräquivalent (Schnee, Eis, Wasser)	L	Double
drySWE	Der gefrorene Anteil am Gesamtschneewasseräquivalent (Schnee, Eis)	L	Double
totDens	Die Gesamtschneedichte	g cm <sup>-3</sup>	Double
dryDens	Die Dichte des gefrorenen Anteils	g cm <sup>-3</sup>	Double
snowDepth	Die Schneehöhe	mm	Double
snowAge	Das Alter der Schneedecke seit dem letzten Neuschnee	d	Double
snowColdContent	Der Kältegehalt der Schneedecke		Double

## Ausgabe

Name	Beschreibung	Einheit	Typ
snowMelt	Die Schneeschmelze	L	Double

## Beschreibung der Komponente

Das Schneemodul folgt dem in *Knauf 1980* beschriebenen Verfahren mit einigen Änderungen und Erweiterungen. Das Schneemodul berücksichtigt die Zustandsänderungen der Schneedecke während ihrer Lebensdauer, insbesondere Dichteänderungen durch Schmelz- und Setzungsvorgänge. Dies ist von Bedeutung, da Schneedecken bis zum Erreichen einer bestimmten Grenzdichte freies Wasser, ähnlich wie ein Schwamm speichern kann und erst nach Überschreiten dieser Grenzdichte eine sprunghafte Wasserabgabe erfolgt. Für die Modellierung werden hierfür unterschiedliche Wasserkapazitäten der Schneedecke betrachtet: Das eigentliche Schneewasseräquivalent (*drySWE*), das der Menge des tatsächlich gefrorenen Wassers entspricht und das Gesamtschneewasseräquivalent (*totSWE*), das zusätzlich noch das flüssige, in der Schneedecke gespeicherte Wasser berücksichtigt. Die Setzung der Schneedecke, die aus dem Eintrag von flüssigem Wasser durch Schneeschmelze an der Oberfläche oder aus Niederschlägen in Form von Regen resultiert, wird nach dem empirischen Schneesetzungsverfahren (snow compaction scheme) von *Bertle 1966* berechnet.

Die Schneedecke und ihre Zustände werden für die Modellierung anhand folgender Parameter beschrieben: Die Schneehöhe (*snowDepth*) [in mm], die Trockenschneedichte (*dryDens*) [in g/cm<sup>3</sup>] als Quotient aus Gesamtwassergehalt und Schneehöhe.

Liegen minimale, mittlere und maximale Lufttemperatur für einen Zeitschritt vor berechnet das Modul getrennte Akkumulations- bzw. Schmelztemperaturen. Hierdurch kann innerhalb eines Zeitschrittes sowohl Akkumulation als auch Schmelze auftreten. Die Berechnung der Temperaturen erfolgen nach:

$$T_{acc} = \frac{T_{min} + T_{avg}}{2} [^{\circ}\text{C}]$$

$$T_{melt} = \frac{T_{max} + T_{avg}}{2} [^{\circ}\text{C}]$$

Ist nur die mittlere Lufttemperatur verfügbar wird anhand dieser entschieden ob sich das Modul in der Akkumulation oder Schmelzphase befindet.



### Akkumulationsphase:

Bei Temperaturen deutlich unter dem dem Gefrierpunkt eine Schneedecke auskühlt. Bevor nun Temperaturen oberhalb des Gefrierpunktes zur Schneeschmelze führen muss zunächst die Temperatur der Schneedecke wieder nahe an die 0 Grad Celsius gebracht werden. Dieser Prozess wird durch die Berechnung eines **Kältegehalt** (*snowColdContent*) der Schneedecke berücksichtigt. Dieser ergibt sich nach:

$$snowColdContent_t = CC_{t-1} + cc\_Factor \times T_{avg} \text{ [mm]}$$

Hierdurch werden negative Lufttemperaturen kumuliert und durch positive Lufttemperaturen und daraus resultierende potentielle Schmelzraten wieder aufgehoben. Erst wenn der Kältegehalt wieder den Wert 0 erreicht kommt es zur Schneeschmelze.

Das **Alter des Schnees** (*snowAge*) in Zeitschritten wird bei jedem Zeitschritt an dem kein Neuschnee fällt aber noch eine Schneedecke vorhanden ist um eins erhöht.

**Schneeakkumulation** tritt dann auf wenn das Modul Niederschlag in fester Form (*netSnow* > 0) erhält. Hier wird dann zunächst die **Neuschneedichte** (*newSnowDens*) in Abhängigkeit von der Lufttemperatur bestimmt. Die Berechnung erfolgt nach *Kuchment et al. 1983* und *Vehviläinen 1992* nach:

$$newSnowDens = 0.13 + 0.0135 \times T_{acc} + 0.00045 \times T_{acc}^2 \text{ [g/cm}^3\text{]}$$

Die aus Schneeniederschlägen (*netSnow*) resultierende **Schneehöhenänderung** ( $\delta SH$ ) berechnet sich dann wie:

$$\delta SH = \frac{netSnow}{newSnowDens} \text{ [mm]}$$

Das **Schneewasseräquivalent** des Vortages (*SWCdry*) erhöht sich damit um den Betrag des Schneeniederschlages nach:

$$SWCdry_t = SWCdry_{t-1} + netSnow \text{ [mm]}$$

Um den gleichen Betrag wird das Trockenschneewasseräquivalent und das Gesamtschneewasseräquivalent erhöht. Handelte es sich beim Niederschlagsereignis um einen Mischniederschlag, wird zusätzlich der Regenanteil dem Gesamtschneewasseräquivalent zugeschlagen.

Ist Regen am Niederschlagsereignis beteiligt, führt dies zu einer Setzung der Schneedecke. Die Berechnung des Setzungsbetrages wird weiter unten dargestellt. Während der Modellierung verhardt die Schneedecke solange in der Akkumulationsphase, bis der vom Anwender festgelegte Temperaturwert für die Schneeschmelze ( $T_{melt}$ ) einen ebenfalls während der Parametrisierung festzulegenden Temperaturgrenzwert (*baseTemp*) überschreitet. Dann tritt sie in die Metamorphosephase, in der die Schmelz- und Setzungsprozesse nachgebildet werden, ein. Sie kann aber jederzeit, bei entsprechend niedrigeren Temperaturen, wieder in die Akkumulationsphase zurückkehren. Durch die unterschiedlichen Temperaturwerte können während eines Zeitschrittes sowohl Akkumulations- als auch Schmelzvorgänge modelliert werden.

### Schmelz- und Setzungsphase:

Überschreitet der vom Anwender bestimmte Temperaturwert ( $T_{melt}$ ) den Temperaturgrenzwert (*baseTemp*), tritt die Schneedecke von der Akkumulations- in die Metamorphosephase über. Die Energie, die dort für die Schneeschmelze zur Verfügung steht, kann auf drei verschiedenen Wegen in die Schneedecke gelangen. Zum ersten durch Eintrag von fühlbarer Wärme durch die Lufttemperatur, zweitens durch Energieeintrag aus Niederschlag in Form von Regen und drittens durch Eintrag auf Grund des Bodenwärmestromes. Aus der Summe der Energieeinträge ergibt sich eine potentielle Schneeschmelzrate ( $Mp$ ). Die Berechnung von  $Mp$  erfolgt nach:

$$Mp = t\_factor \times T_{melt} + r\_factor \times netRain \times T_{melt} + g\_factor \quad [\text{m}^3/\text{s}]$$

$Mp$  wird dann noch an die Hangneigung und die Exposition der räumlichen Modelleinheit angepasst:

$$Mp = Mp \times actSIAsCf \quad [\text{mm}]$$

$Mp$  wird nun zunächst benutzt um den Kältegehalt der Schneedecke wieder auszugleichen und erzeugt erst dann Schneeschmelze. Aus der potentiellen Schmelzrate wird die resultierende maximale Schneehöhenänderung ( $\delta SH$ ) berechnet:

$$\delta SH = \frac{Mp}{dryDens} \quad [\text{mm}]$$

Ist  $\delta SH$  größer als die gesamte Schneehöhe, taut diese komplett ab und das gesamte Schneewasseräquivalent trägt als Schneeschmelze der Abflussbildung bei. Ist dies nicht der Fall wird die Schneehöhe entsprechend reduziert, wobei sich das Schneewasseräquivalent zunächst nicht ändert. Hieraus resultiert eine Zunahme der Gesamtdichte der Schneedecke.

Zusätzlich zu dieser Dichteänderung werden zusätzliche Setzungs- und Dichteänderungen nach dem **snow-compaction-scheme** nach Bertle 1966 berücksichtigt. Diesem Verfahren liegt zugrunde, dass Wasser, gleich ob es aus der temperaturindizierten Schneeschmelze oder aus dem Niederschlag resultiert, in die Schneedecke einsickert und dort zu Setzungsvorgängen durch Umkristallisation des Schnees und durch Strukturänderungen und Verdichtung in der Lagerung führt (Knauf 1980). Die daraus resultierende Setzungsrates wird nach dem in Bertle 1966 beschriebenen Schneesetzungsverfahren berechnet. Diesem Verfahren liegt die Beobachtung eines empirischen Zusammenhangs zwischen zugeführtem freien Wasser und der daraus resultierenden Höhenänderung durch die Setzung zugrunde, der aus Laborversuchen des US-Bureau of Reclamation abgeleitet wurde. Hierfür wird die prozentuale Zunahme des akkumulierten Wassergehaltes im Verhältnis zum Schneewasseräquivalent nach folgender Gleichung berechnet:

$$P_W = \frac{SWC_{tot}}{SWC_{dry}} \times 100 \quad [\%]$$

In dieser Gleichung kommt zum Ausdruck, daß die Schneedeckensetzung um so größer ist, je mehr flüssiges Wasser zugeführt wird (Knauf 1980). Eine Zufuhr von gerade der Wassermenge, die dem Schneewasseräquivalent der Schneedecke entspricht, führt zu einer Halbierung der Schneehöhe durch die Setzung. Die prozentuale Schneehöhenänderung ( $P_H$ ) berechnet sich in Abhängigkeit des zugeführten freien Wassers nach:

$$P_H = 147.4 - 0.474 \times P_W \quad [\%]$$

Daraus ergibt sich die neue Schneehöhe nach:

$$SH = SH \times \frac{P_H}{100} \quad [\text{mm}]$$

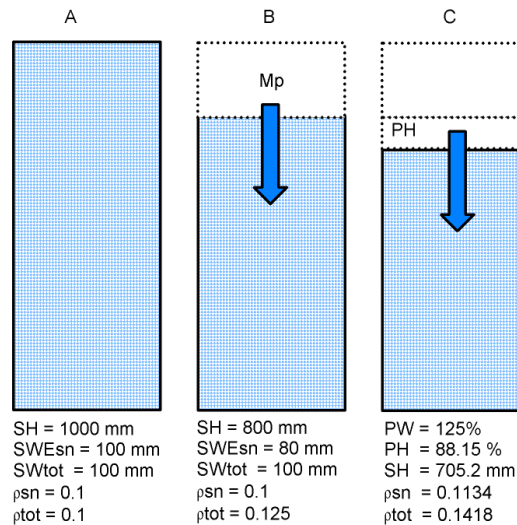
Mit der neu berechneten Schneehöhe wird nun die Gesamtschneedichte ( $totDens$ ) sowie die Trockenschneedichte ( $dryDens$ ) nach folgenden Gleichungen neu berechnet:

$$dryDens = \frac{SWC_{dry}}{SH} \quad [\text{g}/\text{cm}^3]$$

$$totDens = \frac{SWC_{tot}}{SH} \quad [\text{g}/\text{cm}^3]$$

Folgende Abbildung (Abbildung 3) stellt das Snow-Compaction-Scheme noch einmal beispielhaft dar.

**Abbildung 3. Beispielhafte Darstellung der durch Schmelze implizierten Abnahme einer Schneedecke. A: Schneedecke vor Beginn der Schmelze, B: Rückgang durch Schmelze, C: Setzung der Schneedecke durch das Schmelzwasser.**



**Schneeschnelzabfluss:** Wie oben dargestellt ist die Schneedecke in der Lage, bis zu einer gewissen Grenzdichte (*critDens*) freies Wasser in ihren Poren zu speichern. Diese Speicherfähigkeit geht bei Erreichen eines bestimmten Anteils von freiem Wasser im Verhältnis zum Gesamtschneewasseräquivalent (zwischen 40% und 45% nach Bertle 1966, Herrmann 1976, Lang 2005) nahezu vollkommen und irreversibel verloren. Ab diesem Schwellenbereich wird auch ohne Regeneinwirkung die Retentionskapazität einer natürlich entwickelten Schneedecke sprunghaft herabgesetzt, es ist eine ruckhafte Wasserabgabe aus der Schneedecke zu beobachten (Herrmann 1976). Bei der Modellierung wird dieses Verhalten durch die Berechnung eines maximalen Wassergehaltes ( $SWC_{max}$ ) der Schneedecke nach:

$$SWC_{max} = snowCritDens \times SH \text{ [mm]}$$

nachgebildet. Die kritische Grenzdichte (*snowCritDens*) ist dabei vom Anwender anzugeben. Das in der Schneedecke gespeicherte Wasser, das diesen Grenzwert überschreitet, kommt zum Abfluß:

$$Q_{snow} = SWC_{tot} - SWC_{max} \text{ [mm]}$$

In den folgenden Zeitschritten verharrt die Dichte der Schneedecke dann auf der kritischen Grenzdichte, bis sie entweder vollkommen abgetaut ist oder durch erneutes Auftreten von Schneefall wieder in die Akkumulationsphase übergeht.

Für weiterführende Informationen siehe auch: Bertle 1966, Herrmann 1976, Knauf 1980, Kuchment et al. 1983, Lang 2005, Vehviläinen 1992

## JAMContext

Eine Komponente die den allgemeinen JAMS-Kontext beschreibt.

## Metainformationen

Klassifikation	
----------------	--

Paket	jams.model
Autor	Sven Kralisch
Version	1.0_0
Datum	2005-06-22

## Attribute

## Beschreibung der Komponente

Diese Kontextkomponente stellt den allgemeinsten Typ eines JAMS-Kontextes dar. Ihr einziger Zweck ist das Gruppieren von anderen JAMS-Komponenten/-Kontexten, d.h. all ihre Kindkomponenten/-kontexte werden einmal ausgeführt.

## JAMSExecInfo

Komponente zur Darstellung von Simulationsfortschritts und Informationsausgabe

## Metainformationen

Klassifikation	
Paket	jams.components.gui
Autor	Sven Kralisch
Version	1.1_0
Datum	17. June 2006

## Attribute

### Eingabe

Name	Beschreibung	Einheit	Typ
milliSeconds	null		Integer

## Beschreibung der Komponente

Diese Komponente erstellt ein GUI-Panel und zeigt einen Fortschrittsbalken und ein Fenster mit der Ausgabe des Infoprotokolls an. Es wird üblicherweise innerhalb eines zeitlichen oder räumlichen Kontextes verwendet, wobei es den Fortschritt der Kontextiteration anzeigt. Die Anzahl der Iterationen kann mit Hilfe der Variable *counter* definiert werden. Bleibt diese leer, wird stattdessen die Iterationszahl des umgebenden Kontextes benutzt.

## JAMSModel

Eine allgemeine JAMS-Modellkomponente.

## Metainformationen

Klassifikation	
Paket	jams.model
Autor	Sven Kralisch
Version	1.0_0
Datum	2006-05-31

## Attribute

### Eingabe

Name	Beschreibung	Einheit	Typ
workspaceDirectory	das Arbeitsverzeichnis des Modells		DirName

## Beschreibung der Komponente

Diese Komponente dient als ein Top-Level-Container für jedes beliebige JAMS-Modell. Ihr einziger Eingabeparameter ist der relative oder absolute Pfad zum Arbeitsverzeichnis des Modells.

## JAMSSpatialContext

Eine Komponente für die Iteration im Raum.

## Metainformationen

Klassifikation	
Paket	jams.model
Autor	Sven Kralisch
Version	1.0_0
Datum	2005-08-02

## Attribute

### Eingabe

Name	Beschreibung	Einheit	Typ
entities	Liste der räumlichen Modelleinheiten		EntityCollection

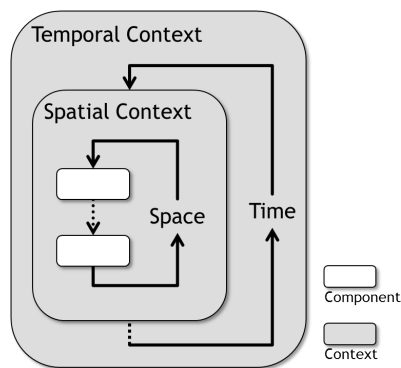
### Ausgabe

Name	Beschreibung	Einheit	Typ
current	null		Entity

## Beschreibung der Komponente

Diese Komponente kann zur Darstellung des Raums in JAMS-Modellen verwendet werden. Auf Grundlage einer Liste räumlicher Einheiten (*entities*) führt es alle seine Kindkomponenten/-kontexte jeweils einmal für jede Einheit aus. Ein Beispiel für einen räumlichen Kontext, der in einen zeitlichen Kontext integriert ist, wird dargestellt in Abbildung 4. Während es über die Liste der Einheiten iteriert, bietet JAMSSpatialContext seinen Kindkomponenten/-kontexten Zugriff auf die Attributwerte der aktuellen Einheit. Über die Methode `getCurrent()` des Attributs *entities* kann während der räumlichen Iteration bei Bedarf auf die aktuelle Modelleinheit zugegriffen werden. Für nähere Informationen zum Datenaustausch durch Kontexte und den Zugriff auf Attribute der Einheit von räumlichen Kontexten beziehen Sie sich bitte auf die Dokumentation von JAMS.

**Abbildung 4. Räumlicher Kontext, integriert in einen zeitlichen Kontext**



## JAMSTemporalContext

Eine Komponente für die Iteration in der Zeit.

## Metainformationen

Klassifikation	
Paket	jams.model
Autor	Sven Kralisch
Version	1.0_1
Datum	2005-07-31

## Attribute

## Eingabe

Name	Beschreibung	Einheit	Typ
timeInterval	Zeitintervall, bestehend aus Startpunkt, Endpunkt und Schrittweite		TimeInterval
printTime	null		Integer

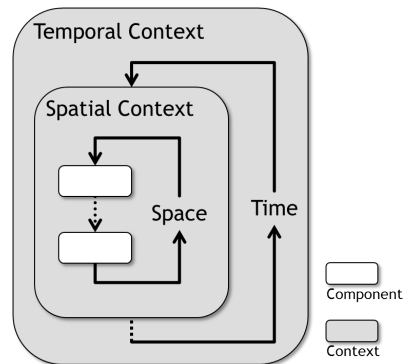
## Ausgabe

Name	Beschreibung	Einheit	Typ
current	aktueller Zeitpunkt		Calendar

## Beschreibung der Komponente

Diese Kontext-Komponente dient zur Iteration über ein definiertes Zeitintervall (Attribut *timeInterval*), welches durch Start- und Endpunkt (zwei Objekte vom Typ `Attribute.Calendar`) sowie Schrittweite (bestehend aus einer Zeiteinheit und Anzahl) gegeben ist. Alle Kindkomponenten/-kontexte von `JAMSTemporalContext` werden einmal für jeden der so definierten diskreten Zeitpunkte ausgeführt. Der aktuelle Zeitpunkt wird im Attribut *current* (Typ `Attribute.Calendar`) bereitgestellt. Zu weiterführenden Informationen über Kontexte siehe auch [ Kralisch et.al 2009 ] bzw. [kralisch\_coupling\_2009]. Ein Beispiel für die Verwendung eines zeitlichen Kontextes ist in Abbildung 5 dargestellt.

Abbildung 5. Zeitlicher Kontext mit eingeschlossenem räumlichen Kontext



## JAMSUnitConverter

### Metainformationen

Klassifikation	
Paket	jams.components.tools
Autor	Sven Kralisch
Version	1.0_0
Datum	1. December 2005

## Attribute

### Eingabe

Name	Beschreibung	Einheit	Typ
inUnit	Unit of input value		String

Name	Beschreibung	Einheit	Typ
outUnit	Unit of output value		String
inValue	Input value		Double[]

## Ausgabe

Name	Beschreibung	Einheit	Typ
outValue	Output value		Double[]

# MapCreator

Eine Komponenten zur Visualisierung eines Einzugsgebietes.

## Metainformationen

Klassifikation	
Paket	jams.components.gui
Autor	Christian Schwartze
Version	1.0_0
Datum	2010-10-22

## Attribute

### Eingabe

Name	Beschreibung	Einheit	Typ
stylesFileName	Name der Datei die den sld Style enthält		String
time	null		Calendar
styleID	ID des Styles der zur Darstellung verwendet werden soll.		Integer
hrus	HRU Entities		EntityCollection
showAttr	Liste der Attribute die dargestellt werden		StringArray
numOfRanges	Anzahl der Intervalle zur Klassifikation der Attribute		StringArray
rangeColor	Farbe zum Ausfüllen der Intervalle		StringArray
shapeFileName1	Name eines Shapefiles zur Darstellung zusätzlicher Layer		String
shapeFileName2	Name eines Shapefiles zur Darstellung zusätzlicher Layer		String
shapeFileName3	Name eines Shapefiles zur Darstellung zusätzlicher Layer		String
shapeColors	Farben für zusätzliche Shapefiles		StringArray



Name	Beschreibung	Einheit	Typ
baseShape	null		String

## Beschreibung der Komponente

Erstellt eine interaktive Visualisierung des Einzugsgebietes. Räumlich ausgeprägte Attribute der HRUs können dargestellt werden und entsprechende Legenden sind verfügbar. Ausserdem stehen grundlegende Funktionen eines typischen Geodaten-Viewers zur Auswahl, z.B. Karte bewegen, Hinein- bzw. Heraus-Zoomen und Selektion. Der Darstellungsstil der zu visualisierenden Attribute kann über eine sld Datei und Farbwerte angepasst werden. Zusätzliche Shape-Layer können hinzu geladen werden. Ein Export-Tool ermöglicht, dass die jeweils dargestellten Attribute (zzgl. weiterer Attribute des Ausgangs-Shapes) in ein neues Shapefile geschrieben werden.

## MapCreator3D

Eine Komponenten für die 3D Visualisierung eines Einzugsgebietes.

## Metainformationen

Klassifikation	
Paket	jams.components.gui
Autor	
Version	
Datum	

## Attribute

### Eingabe

Name	Beschreibung	Einheit	Typ
stylesFileName	Name der Datei die den sld Style enthält		String
styleID	ID des Styles der zur Darstellung verwendet werden soll.		Integer
hrus	Name des HRU Attributes, das dargestellt werden soll		EntityCollection
showAttr	Liste der Attribute die dargestellt werden		StringArray
numOfRanges	Anzahl der Intervalle zur Klassifikation der Attribute		StringArray
rangeColor	Farbe zum Ausfüllen der Intervalle		StringArray
shapeFileName1	Name eines Shapefiles zur Darstellung zusätzlicher Layer		String
shapeFileName2	Name eines Shapefiles zur Darstellung zusätzlicher Layer		String
shapeFileName3	Name eines Shapefiles zur Darstellung zusätzlicher Layer		String

Name	Beschreibung	Einheit	Typ
shapeColors	Farben für zusätzliche Shapefiles		StringArray
baseShape	null		String
heightMap	Dateiname einer Höhenkarte des darzustellenden Einzugsgebietes		String
resolution	Horizontale/Vertikale Auflösung des berechneten 3D Modells		Integer
light	Schalter zur Beleuchtung des 3D Modells		Boolean

## Beschreibung der Komponente

Erstellt eine interaktive dreidimensionale Visualisierung des Einzugsgebietes. Räumlich ausgeprägte Attribute des hrus Attribut können auf die Oberfläche der Darstellung projiziert werden. Der Darstellungsstil kann über eine sld Datei und Farbwerten angepasst werden. Zusätzliche Shape - Layer können hinzu geladen werden.

## PenmanMonteith

Eine Komponente zur Berechnung der potentiellen Landoberflächenverdunstung nach Penman-Monteith. Die Berechnung erfolgt weitestgehend nach .

## Metainformationen

Klassifikation	
Paket	org.unijena.j2k.potET
Autor	Peter Krause
Version	1.0_0
Datum	2011-05-30

## Attribute

### Eingabe

Name	Beschreibung	Einheit	Typ
time	Der aktuelle Zeitschritt		Calendar
tempRes	Die zeitliche Auflösung [h   d   m]		String
wind	Windgeschwindigkeit [m/s]	m/s	Double
tmean	Mittlere Lufttemperatur [°C]	°C	Double
rhum	Relative Luftfeuchtigkeit [%]	%	Double
netRad	Nettostrahlung [MJ/m² tempRes]	MJ m <sup>-2</sup> d <sup>-1</sup>	Double
actRsc0	Stomatawiderstand [s/m]	m s <sup>-1</sup>	Double
elevation	Die Höhe über NN der räumlichen Einheit [m]	m	Double

Name	Beschreibung	Einheit	Typ
area	Die Fläche der räumlichen Einheit [m²]	m²	Double
actLAI	Der Blattflächenindex [-]		Double
actEffH	Die effektive Wuchshöhe der Vegetation	m	Double

## Ausgabe

Name	Beschreibung	Einheit	Typ
potET	Die potentielle Verdunstung [mm/ tempRes]	mm d <sup>-1</sup>	Double
actET	Die aktuelle Verdunstung [mm/ tempRes]	mm d <sup>-1</sup>	Double
ea	null	hPa	Double
rs	Der Verdunstungswiderstand der Vegetation [s/m]	s m <sup>-1</sup>	Double
ra	Der aerodynamische Widerstand des Bestandes [s/m]	s m <sup>-1</sup>	Double

## Beschreibung der Komponente

Mit diesem Modul wird die potentielle Verdunstung nach Penman-Monteith ( *Monteith 1975* ) für verschiedene Landbedeckungen berechnet. Die Berechnung der Bestandverdunstung erfolgt in mehreren Schritten weitestgehend nach [ Allen et.al 1998 ] unter Einbeziehung einer Vielzahl von Parametern. Die Berechnung der potentiellen Verdunstung erfolgt nach:

$$ETP = \frac{1}{L} \times \frac{s \times (R_N - G) + \rho \times c_p \times \frac{es - ea}{ra}}{s + \gamma \times \left(1 + \frac{rs}{ra}\right)}$$

mit:

$L$ ... Latente Verdunstungswärme [MJ/kg/ZS]

$s$ ... Steigung der Sättigungsdampfdruckkurve [hPa/K]

$R_N$  ... Nettostrahlung [MJ/m²/ZS]

$G$  ... Bodenwärmestrom [MJ/m²/ZS]

$\rho$  ... Dichte der Luft [kg/m³]

$c_p$ ... Spezifische Wärmekapazität der der Luft bei konstantem Druck [J/kg/K]

$es$  ... Sättigungsdampfdruck [kPa]

$ea$ ... Aktueller Dampfdruck [hPa]

$ra$  ... Aerodynamischer Widerstand der Bodenbedeckung [s/m]

$\gamma$ ... Psychrometerkonstante [hPa/K]

$rs$ ... Oberflächenwiderstand der Bodenbeckung [s/m]

Die latente Verdunstungswärme ( $L$ ) berechnet sich aus der mittleren Lufttemperatur  $T_{avg}$  in °C nach:

$$L = \frac{25001 - (2.361 \times T_{avg})}{1000} \text{ [MJ/kg/ZS]}$$

Der Sättigungsdampfdruck ( $es$ ) in kPa der Luft bei der Temperatur ( $T_{avg}$ ) ergibt sich als:

$$es = 0.6108 \times e^{\frac{17.27 \times T_{avg}}{237.3 + T_{avg}}} \text{ [kPa]}$$

Der aktuelle Dampfdruck ( $ea$ ) ergibt sich aus dem Sättigungsdampfdruck und der relativen Luftfeuchte (%) nach:

$$ea = es \times \frac{U}{100} \text{ [kPa]}$$

Aus dem Sättigungsdampfdruck ( $es$ ) und der Lufttemperatur ( $T_{avg}$ ) berechnet sich die Steigung der Sättigungsdampfdruckkurve ( $s$ ) nach

$$s = es \times \frac{4098}{(T_{avg} + 237.3)^2} \text{ [kPa/°C]}$$

Der Luftdruck ( $P_z$ ) in der Höhe ( $z$ ) wird aus der umgestellten barometrischen Höhenformel ermittelt:

$$P_z = 101.3 \times e^{-1 \times \frac{9.811}{8314.3 \times T_{abs}}} \times z \text{ [kPa]}$$

mit:

$p_0$ ... Luftdruck auf Meeresniveau (= 101.3) [kPa]

$g$ ... Erdbeschleunigung (= 9.811) [m/s]

$R$ ... Gaskonstante (= 8314.3) [Jkmol/K]

$T_{abs}$ ... absolute Lufttemperatur [K]

Die Psychrometerkonstante ( $\gamma$ ) ergibt sich aus der spezifischen Wärmekapazität der Luft ( $= 1.013 \times 10^{-3}$ ), dem Luftdruck  $P_z$ , dem Verhältnis der Molgewichte von trockener Luft und Wasserdampf ( $= 0.622$ ) und der latenten Verdunstungswärme ( $L$ ) nach:

$$\gamma = \frac{1.013 \times 10^{-3} \times P_z}{0.622 \times L} \text{ [kPa/°C]}$$

### **Berechnung bestandsspezifischer Parameter**

Der Einfluss verschiedener Vegetationsformen auf die Verdunstung wird im Penman-Monteith-Ansatz durch zwei verschiedene Widerstände berücksichtigt, dem Oberflächenwiderstand ( $rs$ ) und dem aerodynamischen Widerstand ( $ra$ ). Für die Berechnung der Widerstände werden landnutzungsspezifische Parameter benötigt.

Im einzelnen sind dies:

- der Blattflächenindex LAI
- die effektive Bewuchshöhe (eff.Bh.)

- und die Oberflächenwiderstände bei Wassersättigung

Deren Werte sind für verschiedene Bodenbedeckungsklassen in folgender Abbildung (Abbildung 6) dargestellt:

**Abbildung 6. Landnutzungsparameter verschiedener Bodenbedeckungsklassen**

Landnutzung	Albedo	minimaler Oberflächenwiderstand bei wassergesättigtem Boden $r_{sc_0}$						LAI		eff. Bh.		Wurzelteiefe
	[-]	[ $\text{sm}^{-1}$ ]						[-]		[m]		[dm]
		Jan. Jul.	Feb. Aug.	Mär. Sep.	Apr. Okt.	Mai Nov.	Jun. Dez.	d1 d3	d2 d4	d1 d3	d2 d4	
Siedlungsfl. VG > 80%	0,10	100 100	100 100	100 100	100 100	100 100	100 100	1 1	1 1	10 10	10 10	2
Siedlungsfl. VG < 80%	0,10	90 55	90 55	80 60	70 70	50 90	55 90	2 5	5 2	3 5	5 3	3
Wiesen und Weiden	0,25	80 45	80 45	70 50	60 60	40 80	45 80	2 4	4 2	0,15 0,3	0,4 0,15	4
Nadelwald	0,12	70 45	70 45	60 50	55 65	45 70	45 70	8 12	12 8	10 10	10 10	12
Laubwald	0,17	80 55	80 55	70 60	65 75	55 80	55 80	0,5 8	8 0,5	0,3 10	10 0,3	14
Mischwald	0,15	75 50	75 50	65 55	60 70	50 75	50 75	2 10	10 2	3 10	10 3	13
Ackerland	0,25	80 50	80 50	75 50	65 65	45 80	50 80	1 3	5 1	0,05 0,3	0,5 0,05	4
Strauchvegetation	0,20	80 50	80 55	70 55	60 70	50 80	50 80	3 5	5 3	1,5 2,5	2,5 1,5	5
Moore und Feuchflächen	0,20	80 50	80 55	70 55	60 70	50 80	50 80	2 5	5 2	3 5	5 3	5
Freiflächen	0,10	150 150	150 150	150 150	150 150	150 150	150 150	1 1	1 1	0,05 0,05	0,05 0,05	1
Wasserflächen	0,05	20 20	20 20	20 20	20 20	20 20	20 20	1 1	1 1	0,1 0,1	0,1 0,1	0

Weiterhin sind die bestandsspezifischen Albedowerte enthalten, die bei der Berechnung der Strahlungsbilanz eingesetzt werden. Der Blattflächenindex und die effektive Bewuchshöhe sind in Form von markanten Stellen ( $d1...d4$ ) des Jahresgangs dargestellt. Die Punkte repräsentieren den Beginn der Vegetationsphase ( $d1$ ), das Erreichen der maximalen Ausprägung oder Vollreife ( $d2$ ), die Vollreifephase bis zum Punkt  $d3$  und dann die Abnahme bis zum Ende der Vegetationsperiode ( $d4$ ). Die einzelnen Punkte werden durch die julianischen Tageswerte ( $d1 = 110$ ,  $d2 = 150$ ,  $d3 = 250$ ,  $d4 = 280$ ) für Gebiete in ca. 400 m Höhe repräsentiert. Für andere Höhen ( $z$ ) werden diese Punkte nach folgender empirischen Beziehung approximiert:

$$d_1(z) = d_1(400) + 0.025 \times (z - 400)$$

$$d_2(z) = d_2(400) + 0.025 \times (z - 400)$$

$$d_3(z) = d_3(400) + 0.025 \times (z - 400)$$

$$d_4(z) = d_4(400) + 0.025 \times (z - 400)$$

Die Werte zwischen den einzelnen Punkten werden linear interpoliert. Der aerodynamische Widerstand ( $ra$ ) für Vegetation mit einer effektiven Bewuchshöhe von weniger als 10 m der jeweiligen Landnutzungsart lässt sich nach folgender Gleichung berechnen:

$$ra = \frac{9.5}{v^2} \times \left( \log \frac{2}{z_0} \right)^2 \text{ [s/m]}$$

mit

$z_0$ ... aerodynamische Rauigkeitslänge (# 0.1\*effektive Bewuchshöhe) [m]

$v_2$  ... Windgeschwindigkeit in 2 m Höhe [m/s]

Für effektive Bewuchshöhen von gleich oder mehr als 10 m berechnet sich die aerodynamische Widerstand vereinfacht nach:

$$r_a = \frac{20}{0.41^2 \times v_2} \text{ [s/m]}$$

Der Oberflächenwiderstand der jeweiligen Landbedeckung berechnet sich nach:

$$r_s = \left( \frac{1-A}{r_{sc}} + \frac{A}{r_{ss}} \right)^{-1} \text{ [s/m]}$$

mit

$r_{sc}$ ... Oberflächenwiderstand [s/m]

$A$ ... 0.7LAI [-]

$r_{ss}$ ... Oberflächenwiderstand von unbewachsenem Boden [s/m]

Zu weiterführenden Informationen siehe auch: *Allen et.al 1998* , *Monteith 1975*

## RainCorrectionRichter

Eine Komponente zur Korrektur von Niederschlagsmesswerten nach dem Verfahren von .

## Metainformationen

Klassifikation	
Paket	org.unijena.j2k.inputData
Autor	Peter Krause
Version	1.0_0
Datum	2011-05-30

## Attribute

## Eingabe

Name	Beschreibung	Einheit	Typ
time	der aktuelle Modellierungszeitpunkt		Calendar
precip	die unkorrigierten Niederschlagswerte	mm	DoubleArray
temperature	Lufttemperaturwerte der Niederschlagsstationen oder von umliegenden Klimastationen	°C	DoubleArray

Name	Beschreibung	Einheit	Typ
tempElevation	die Stationshöhen für die Temperaturmessstationen	m	DoubleArray
tempXCoord	die Rechtswerte für die Temperaturmessstationen		DoubleArray
tempYCoord	die Hochwerte für die Temperaturmessstationen		DoubleArray
tempRegCoeff	Achsenabschnitt, Steigung und Bestimmtheitsmaß aus der Regression zwischen Temperaturwerten und Stationshöhen		DoubleArray
rainElevation	die Stationshöhen für die Niederschlagsstationen	m	DoubleArray
rainXCoord	die Rechtswerte für die Niederschlagsstationen		DoubleArray
rainYCoord	die Hochwerte für die Niederschlagsstationen		DoubleArray
tempNIDW	Anzahl von Temperaturstationen die bei der Interpolation berücksichtigt werden sollen		Integer
pIDW	die Potenz für die Gewichtungsberechnung der Interpolation der Temperaturwerte		Double
regThres	der Grenzwert für die Höhenkorrektur		Double
snow_trs	Lufttemperatur bei der 50% des Niederschlags als Regen und 50% als Schnee fallen	°C	Double
snow_trans	Temperaturintervall um snow_trs zur Bestimmung der Lufttemperaturen für Schnee- oder Regenniederschläge	K	Double

## Ausgabe

Name	Beschreibung	Einheit	Typ
rcorr	die korrigierten Niederschlagswerte	mm	DoubleArray

## Beschreibung der Komponente

Mit dieser Komponente werden Niederschlagsmesswerte nach dem Verfahren von [ Richter 1995 ] hinsichtlich ihrer systematischen Messfehler korrigiert. Hierzu wird zunächst für jede Niederschlagsstation eine Temperatur bestimmt. Liegt diese nicht direkt an der Niederschlagsstation vor, wird sie aus umliegenden Stationen mit einer Kombination zwischen Inverser Distanz Interpolation und optionaler Höhenkorrektur ermittelt. Auf Basis der Lufttemperatur wird der Niederschlag dann in Regen und Schnee aufgeteilt.

## Korrektur des Benetzungs- und Verdunstungsfehlers

Die Korrektur des Benetzungs- und Verdunstungsfehlers erfolgt nach Untersuchungen an Hellmann-Niederschlagsmessern von [ Richter 1995 ]. Um eine stetige Korrektur des Fehlers, der aus dem Benetzungs- und Verdunstungsverlust resultiert, zu ermöglichen, wurden für das Modellsystem J2000 logarithmische Funktionen separat für das Sommer- (Mai - Oktober) und Winterhalbjahr (November - April) an die diskreten, tabellierten Werte approximiert. Übersteigt die Niederschlagshöhe den Wert von 9 mm, wird der Benetzungs- und Verdunstungsfehler auf einen konstanten Wert gesetzt.

Für Niederschlagshöhen #9.0 mm berechnet sich der Benetzungs- und Verdunstungsfehler nach:

$$BV_{Som} = 0.08 \times \ln N + 0.225 \text{ [mm]}$$

$$BV_{Win} = 0.08 \times \ln N + 0.13 \text{ [mm]}$$

Für Niederschlagshöhen >9.0 mm beträgt der Benetzungs- und Verdunstungsfehler:

$$BV_{Som} = 0.47 \text{ [mm]}$$

$$BV_{Win} = 0.30 \text{ [mm]}$$

## Korrektur des Windfehlers

Die Quantifizierung des zu erwartenden Niederschlagsfehlers erfolgt nach Untersuchungen von [ Richter 1995 ] als Funktion der Niederschlagshöhe und der Stationslage. Es wird angenommen, dass sich der relative Windfehler ( $KR_{Wind}$ ) für sowohl Regen- als auch Schneeniederschläge deutlich umgekehrt proportional zu den Niederschlagshöhen ( $P_m$ ) verhält. Die Berechnung erfolgt nach folgenden Gleichungen:

$$KR_{Wind} = \begin{cases} 0.1349 \times P_m^{-0.949} & \text{für } T_{mean} > T_{crit} \\ 0.5319 \times P_m^{-0.197} & \text{für } T_{mean} \leq T_{crit} \end{cases} [-]$$

Die Berechnung der um Verdunstungs- und Windfehler korrigierten Niederschlagshöhe erfolgt schließlich nach:

$$P_{kor} = P_m + P_m \times KR_{Wind} + BV_{Som} + BV_{Win} [\text{mmd}^{-1}]$$

## Regionalisation

Eine Komponente, die aktuelle Werte für den jeweiligen Modellzeitschritt zur Verfügung stellt.

## Metainformationen

Klassifikation	
Paket	org.unijena.j2k.regionalisation
Autor	
Version	
Datum	

## Attribute

### Eingabe

Name	Beschreibung	Einheit	Typ
dataArray	Liste mit Eingangsdaten (Stationswerten) die regionalisiert werden sollen		DoubleArray
regCoeff	Koeffizienten (Achsenabschnitt, Steigung, Bestimmtheitsmaß) der linearen Regression		DoubleArray



Name	Beschreibung	Einheit	Typ
	zwischen Datenwerten und Höhenlage der Stationen		
statElevation	Liste die Höhenangabe für die Stationen an denen die Datenwerte aufgenommen wurden enthält		DoubleArray
statWeights	null		DoubleArray
statOrder	nach Gewicht geordnete Liste von Arrayindizes zur Identifizierung der Stationen		IntegerArray
entityElevation	die Höhe der aktuellen Modellentität		Double
nidw	Anzahl der Stationen, die bei der Interpolation berücksichtigt werden sollen		Integer
elevationCorrection	Schalter zum Ein- oder Ausschalten der optionalen Höhenkorrektur		Boolean
rsqThreshold	Grenzwert für das Bestimmtheitsmaß der linearen Regression, der für die Aktivierung der Höhenkorrektur überschritten werden muss		Double
fixedMinimum	absolut niedrigster Wert, der nicht unterschritten werden darf		Double
fixedMaximum	null		Double

## Ausgabe

Name	Beschreibung	Einheit	Typ
dataValue	der regionalisierte Datenwert für den aktuellen Zeitschritt und die aktuelle Modellentität		Double

## Beschreibung der Komponente

Diese Komponente stellt aktuelle Werte für den jeweiligen Modellzeitschritt zur Verfügung. Hierzu erhält sie Reihen von zeitlich variablen Werten, die von anderen Komponenten für Standardjahre berechnet werden. Diese Reihen besitzen je nach zeitlicher Auflösung 12 (monatlich), 366 (täglich) oder 8784 (stündlich) Elemente.

## Regionalisierung von Klima- und Niederschlagsdaten

Allgemeiner Verfahrensgang

**1. Berechnung der linearen Regression zwischen den täglichen Stationsmesswerten und den Stationshöhen.** Dabei wird das Bestimmtheitsmaß ( $r^2$ ) und die Steigung der Regressionsgeraden ( $b_H$ ) dieser Beziehung berechnet. Es wird angenommen, dass der Messwert ( $MW$ ) linear von der Geländehöhe ( $H$ ) abhängt, nach:

$$MW = a_H + b_H \times H$$

Die Unbekannten  $a_H$  und  $b_H$  werden nach der Gaußschen Methode der kleinsten Quadrate bestimmt:

$$b_H = \frac{\sum_{i=1}^n (H_i - \bar{H}) \times (MW_i - \bar{MW})}{\sum_{i=1}^n (H_i - \bar{H})^2}$$

$$a_H = \bar{MW} - b_H \times \bar{H}$$

Der Korrelationskoeffizient der Regression berechnet sich nach:

$$r = \frac{\sum_{i=1}^n (H_i - \bar{H}) \times (MW_i - \bar{MW})}{\sqrt{\sum_{i=1}^n (H_i - \bar{H})^2 \times \sum_{i=1}^n (MW_i - \bar{MW})^2}}$$

**2. Bestimmung der n Messstationen, die der jeweiligen HRU am nächsten liegen.** Die Zahl n, die während der Parametrisierung des Modells angegeben werden muss, ist von der Dichte des Stationsmessnetzes und der Lage der einzelnen Stationen abhängig.

Für jeden Datensatz muss im Vorfeld bestimmt werden, wie viele Stationen (n) zur Regionalisierung herangezogen werden sollen. Weiter ist ein Wichtungsfaktor (pIDW) anzugeben. Anhand der Rechts- und Hochwerte aller Stationen und der Koordinaten der betreffenden HRU werden die n-nächsten Stationen nach folgender Berechnungsvorschrift bestimmt. Der erste Schritt ist die Berechnung der Entfernung ( $Dist(i)$ ) jeder Stationen zur betrachteten Fläche nach:

$$Dist(i) = \sqrt{(RW_{stat(i)} - RW_{DF})^2 + (HW_{stat(i)} - HW_{DF})^2}$$

mit

$RW$  ... Rechtswert der Station i...n, bzw. der HRU (DF)

$HW$  ... Hochwert der Station i...n, bzw. der HRU (DF)

Aus den so ermittelten Entfernungen werden die n Stationen mit den geringsten Entfernungen zur jeweiligen HRU für die weiteren Berechnungen herangezogen. Die Entfernungen dieser Stationen werden durch Potenzierung mit dem Wichtungsfaktor pIDW zu gewichteten Entfernungen ( $\omega Dist(i)$ ) umgerechnet. Mit diesem Wichtungsfaktor kann der Einfluss von naheliegenden Stationen verstärkt und der von weiter entfernt liegenden abgeschwächt werden. Gute Ergebnisse werden mit Werten von 2 oder 3 für pIDW erzielt.

**3. Mit einem Inverse-Distance-Weighted Verfahren (IDW)** werden die Gewichte der n Stationen in Abhängigkeit von ihrer Entfernung für jede HRU in einem anderen Modul bestimmt. Durch das IDW-Verfahren wird die horizontale Variabilität der Stationsdaten, entsprechend ihrer Lage im Raum, berücksichtigt. Die Berechnung erfolgt nach:

$$W(i) = \frac{\left( \frac{\sum_{i=1}^n \omega Dist(i)}{\omega Dist(i)} \right)}{\sum_{i=1}^n \left( \frac{\sum_{i=1}^n \omega Dist(i)}{\omega Dist(i)} \right)}$$

**4. Berechnung des Datenwertes für jede HRU** mit den Gewichten aus Punkt 3 und einer optionalen Höhenkorrektur, zur Berücksichtigung der vertikalen Variabilität. Die Höhenkorrektur wird nur dann durchgeführt, wenn das unter 1. berechnete Bestimmtheitsmaß einen, vom Anwender anzugebenden, Grenzwert übersteigt. Die Berechnung ohne die optionale Höhenkorrektur erfolgt nach:

$$DW_{DF} = \sum_{i=1}^n MW(i) \times W(i)$$

Bei Datenwerten, die bekanntermaßen einen Höheneffekt aufweisen, werden die Messwerte bei genügend enger Regressionsbeziehung ( $r^2$  größer als ein vom Anwender anzugebender Grenzwert) noch zusätzlich höhenkorrigiert. Die Berechnung erfolgt dann nach:

$$DW_{DF} = \sum_{i=1}^n ((\Delta H(i) \times b_H + MW(i)) \times W(i))$$

mit  $\Delta H(i)$  ... Höhendifferenz zwischen der Station  $i$  und der HRU

$b_H$ ... Steigung der Regressionsgeraden

## StandardEfficiencyCalculator

Eine KDataReaderomponente zum Beurteilen der Modellqualität.

### Metainformationen

Klassifikation	
Paket	org.unijena.j2k.efficiencies
Autor	Peter Krause
Version	1.0_0
Datum	2010-10-29

### Attribute

#### Eingabe

Name	Beschreibung	Einheit	Typ
time	der aktuelle Modellzeitschritt		Calendar
modelTimeInterval	das Modellierungszeitintervall		TimeInterval
effTimeInterval	Zeitintervall für das die Gütemaße berechnet werden sollen		TimeInterval
effMonthList	eine kommaseparierte Liste von Monaten (1-12) für die die Güte berechnet werden soll. Diese Variable kann ausgelassen werden wenn sie nicht benötigt wird.		IntegerArray
effMethod	eine kommaseparierte Liste von Integerwerten, die entsprechenden Gütekriterien entsprechen		IntegerArray
prediction	null		Double
validation	der Vergleichswert gegen den die Prüfung stattfinden soll		Double

#### Ausgabe

Name	Beschreibung	Einheit	Typ
e1	die Nash-Sutcliffe Effizienz mit Potenz 1	n/a	Double

Name	Beschreibung	Einheit	Typ
e2	die Nash-Sutcliffe Effizienz mit Potenz 2	n/a	Double
le1	die Nash-Sutcliffe Effizienz berechnet aus logarithmierten Werten mit Potenz 1	n/a	Double
le2	die Nash-Sutcliffe Effizienz berechnet aus logarithmierten Werten mit Potenz 2	n/a	Double
ioa1	der Index of Agreement mit Potenz 1	n/a	Double
ioa2	der Index of Agreement mit Potenz 2	n/a	Double
rsq	das Bestimmtheitsmaß	n/a	Double
grad	die Steigung der linearen Ausgleichsgeraden	n/a	Double
wrsq	eine gewichtete Form des Bestimmtheitsmaßes	n/a	Double
dsGrad	null	n/a	Double
absVolErr	null	same as values	Double
rmse	der mittlere quadratische Fehler	same as values	Double
pbias	der mittlere prozentuale Fehler	%	Double
apbias	der mittlere absolute prozentuale Fehler	%	Double
predictionValues	ein Array, das sämtliche Simulationswerte speichert		DoubleArray

## Beschreibung der Komponente

Mit dieser Komponente können eine Anzahl von Gütemaßen zur Beurteilung der Modellqualität anhand gemessener Werte berechnet werden. Hierzu muss der Komponente der Simulationswert (*prediction*) und ein Vergleichswert (*validation*) für jeden Zeitschritt der Modelllaufzeit übergeben werden. Zusätzlich benötigt die Komponente zwei Zeitintervalle, wobei das erste das Modellzeitintervall (*modelTimeInterval*) ist, das zweite dann ein beliebiger Zeitraum (*effTimeInterval*) innerhalb des Modellzeitintervalls sein darf. Hierdurch, kann beispielsweise eine „warming up“ Periode zu Beginn der Modelllaufzeit aus der Effizienzberechnung ausgeschlossen werden oder Gütemaße für einzelne Jahre bestimmt werden. Es kann auch eine Liste von Monaten (1-12) in der Variablen „*effMonthList*“ angegeben werden um kontinuierlich nur einzelne Monate auszuwerten.

Während der Modelllaufzeit speichert die Komponente lediglich die einlaufenden Einzelwerte und legt diese in entsprechenden Strukturen ab. Fehlende Werte werden als spezieller Fehlwert gespeichert. Während der „cleanUp“ Phase erfolgt dann die Berechnung der spezifischen Gütemaße, die weiter unten beschrieben werden. Diese können durch Angabe von Identifikationsnummern in der Variable „*effMethod*“ ausgewählt werden. Im Einzelnen sind dies:

- 0 - für die Nash-Sutcliffe Effizienz mit Potenz 1
- 1 – für die Nash-Sutcliffe Effizienz mit Potenz 2
- 3 – für die Nash-Sutcliffe Effizienz berechnet aus logarithmierten Werten mit Potenz 1
- 4 – für die Nash-Sutcliffe Effizienz berechnet aus logarithmierten Werten mit Potenz 2
- 5 – der Index of Agreement mit Potenz 1

- 6 – der Index of Agreement mit Potenz 2
- 7 – das Bestimmtheitsmaß
- 8 – eine gewichtete Form des Bestimmtheitsmaßes nach [ Krause et al. 2005 ]
- 9 – die Steigung der Regressionsgerade aus der Doppelsummenanalyse
- 10 – der absolute, kumulierte Volumenfehler
- 11 – der mittlere quadratische Fehler
- 12 – der mittlere prozentuale Fehler
- 13 – der mittlere absolute prozentuale Fehler

Die Berechnung der einzelnen Gütemaße, mit dem Simulationswert  $p_i$  und dem Vergleichswert  $o_i$  erfolgt nach:

Nash-Sutcliffe Effizienz mit Potenz 1 ( $E1$ ):

$$E1 = 1 - \frac{\sum_{i=1}^n |p_i - o_i|}{\sum_{i=1}^n |o_i - \bar{o}|}$$

Nash-Sutcliffe Effizienz mit Potenz 2 ( $E2$ ):

$$E2 = 1 - \frac{\sum_{i=1}^n (p_i - o_i)^2}{\sum_{i=1}^n (o_i - \bar{o})^2}$$

Nash-Sutcliffe Effizienz aus logarithmierten Werten mit Potenz 1 ( $\ln E1$ ):

$$\ln E1 = 1 - \frac{\sum_{i=1}^n |\ln p_i - \ln o_i|}{\sum_{i=1}^n |\ln o_i - \ln \bar{o}|}$$

Nash-Sutcliffe Effizienz aus logarithmierten Werten mit Potenz 2 ( $\ln E2$ ):

$$\ln E2 = 1 - \frac{\sum_{i=1}^n (\ln p_i - \ln o_i)^2}{\sum_{i=1}^n (\ln o_i - \ln \bar{o})^2}$$

Bei den beiden Formen, die aus logarithmierten Werten berechnet werden besteht das Problem wenn einer oder beide Werten 0 zeigen, da hierfür der Logarithmus nicht definiert ist. Hier werden intern folgende Fälle unterschieden:

- Ist einer der beiden Werte 0 so wird das Wertepaar bei der Berechnung nicht berücksichtigt.
- Sind beide Werte 0 so wird eine einfache oder quadratische Differenz von ebenfalls 0 angenommen.

Index of Agreement mit Potenz 1 ( $ioa1$ ):

$$ioa1 = 1 - \frac{\sum_{i=1}^n |p_i - o_i|}{\sum_{i=1}^n |(p_i - \bar{o}) + (o_i - \bar{o})|}$$

Index of Agreement mit Potenz 2 ( $ioa2$ ):

$$ioa2 = 1 - \frac{\sum_{i=1}^n (p_i - o_i)^2}{\sum_{i=1}^n ((p_i - \bar{o}) + (o_i - \bar{o}))^2}$$

Das Bestimmtheitsmaß (*rsq*):

$$rsq = \left( \frac{\sum_{i=1}^n (o_i - \bar{o}) \times (p_i - \bar{p})}{\sqrt{\sum_{i=1}^n (o_i - \bar{o})^2} \times \sqrt{\sum_{i=1}^n (p_i - \bar{p})^2}} \right)^2$$

Eine gewichtete Form des Bestimmtheitsmaßes mit der Steigung der Regressionsgeraden (*b*) als Gewicht (*wrsq*):

$$wrsq = \begin{cases} |b| \times rsq & \text{für } b \leq 1 \\ |b|^{-1} \times rsq & \text{für } b > 1 \end{cases}$$

Die Steigung der linearen Ausgleichsgeraden der kumulierten Simulations- und Vergleichswerten (*dsgrad*) (Abbildung 7, „dsgrad“):

### Abbildung 7. dsgrad

Der absolute, kumulierte Volumenfehler (*ave*):

$$ave = \sum_{i=1}^n |p_i - o_i|$$

Der mittlere quadratische Fehler (*rmse*):

$$rmse = \sqrt{\frac{1}{n} \sum_{i=1}^n (p_i - o_i)^2}$$

Der mittlere prozentuale Fehler (*pbias*):

$$pbias = \frac{\sum_{i=1}^n (p_i - o_i)}{\sum_{i=1}^n o_i} \times 100$$

Der mittlere absolute prozentuale Fehler (*apbias*):

$$apbias = \frac{\sum_{i=1}^n |p_i - o_i|}{\sum_{i=1}^n o_i} \times 100$$

# StandardEntityReaderUpstreamTopo

## Metainformationen

Klassifikation	
----------------	--

Paket	io
Autor	Sven Kralisch
Version	1.0
Datum	2015-07-08

## Attribute

### Eingabe

Name	Beschreibung	Einheit	Typ
upStreamCatchmentAttribute	Name of the attribute describing the upstream catchment relation		String
hruFileName	HRU parameter file name		String
reachFileName	Reach parameter file name		String
hruIDAttribute	Name of the attribute containing the HRU identifiers		String
reachIDAttribute	Name of the attribute containing the reach identifiers		String
hru2hruAttribute	Name of the attribute describing the HRU to HRU relation in the input file		String
hru2reachAttribute	Name of the attribute describing the HRU to reach relation in the input file		String
reach2reachAttribute	Name of the attribute describing the reach to reach relation in the input file		String
subcatchmentReachID	ID of a reach defining a sub-catchment. Only hrus/reaches draining to this reach will be used on the resulting entity collections		Double

### Ausgabe

Name	Beschreibung	Einheit	Typ
hrus	Collection of hru objects		EntityCollection
reaches	Collection of reach objects		EntityCollection

## StandardGroundwaterParaReader

Eine Komponente zum Auslesen von hydrogeologischen Daten aus einer ASCII-Datei.

### Metainformationen

Klassifikation	
Paket	org.unijena.j2k.io

Autor	Sven Kralisch
Version	1.1_0
Datum	2005-11-10

## Attribute

### Eingabe

Name	Beschreibung	Einheit	Typ
gwFileName	Name der hydrogeologischen Parameterdatei		String
hrus	Liste der HRU-Objekte		EntityCollection

## Beschreibung der Komponente

Diese Komponente liest hydrogeologische Daten aus einer ASCII-Datei, die durch *gwFileName* gekennzeichnet ist. Für jede Einheit Liste *hrus* wählt es den hydrogeologischen Typen, der durch sein Attribut *hgeoID* identifiziert wird, und fügt entsprechende hydrogeologische Informationen zum Attributeset der Einheit. Ein Beispiel für ein hydrogeologisches Parameterfile wird in folgender Abbildung (Abbildung 8) dargestellt.

**Abbildung 8. Beispiel für eine ASCII-Datei mit hydrogeologischen Daten**

	A	B	C	D	E	F
1	#hgeo.par					
2	GID	RG1_max	RG2_max	RG1_k	RG2_k	RG1_active
3	1	0	0	0	0	0
4	-9999	-9999	-9999	-9999	-9999	1
5	n/a	mm	mm	d	d	n/a
6	1	90	220	8	50	1
7	2	90	1050	12	572	1
8	3	80	825	14	677	1
9	4	70	750	16	671	1

## StandardLUReader

Eine Komponente zum Auslesen von Landnutzungsinformation aus einem ASCII-File.

### Metainformationen

Klassifikation	
Paket	org.unijena.j2k.io
Autor	Sven Kralisch
Version	1.1_0
Datum	2005-11-10



## Attribute

### Eingabe

Name	Beschreibung	Einheit	Typ
luFileName	Dateiname von Landnutzungsparametern		String
hrus	Liste von HRU-Objekten		EntityCollection

## Beschreibung der Komponente

Diese Komponente liest Landnutzungsinformation aus einer ASCII-Datei, die durch *luFileName* identifiziert wird. Für jede Einheit der Liste *hrus* wählt es den Landnutzungstypen, der durch sein Attribut *landuseID* identifiziert wird, und fügt die entsprechende Landnutzungsinformation zum Attributeset der Einheit hinzu. Ein Beispiel für ein Parameterfile der Landnutzung wird in der Abbildung (Abbildung 9) dargestellt.

**Abbildung 9. Ausschnitt aus einem Beispiel einer ASCII-Datei mit Landnutzungsinformation**

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	# Parameter File created: 04.12.2002													
2	LID	albedo	RSC0_1	RSC0_2	RSC0_3	RSC0_4	RSC0_5	RSC0_6	RSC0_7	RSC0_8	RSC0_9	RSC0_10	RSC0_11	RSC0_12
3	1	0	0	0	0	0	0	0	0	0	0	0	0	0
4	-9999	100	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
5	n/a	%	s/m	s/m	s/m	s/m	s/m	s/m	s/m	s/m	s/m	s/m	s/m	s/m
6	1	0.1	100	100	100	100	100	100	100	100	100	100	100	100
7	2	0.1	90	90	80	70	50	55	55	55	60	70	90	90
8	3	0.25	80	80	70	60	40	45	45	45	50	60	80	80
9	4	0.12	70	70	60	55	45	45	45	45	50	65	70	70
10	5	0.17	80	80	70	65	55	55	55	55	60	75	80	80
11	6	0.15	75	75	65	60	50	50	50	50	55	70	75	75
12	7	0.25	80	80	75	65	45	50	50	50	50	65	80	80
13	8	0.2	80	80	70	60	50	50	50	55	55	70	80	80
14	9	0.2	80	80	70	60	50	50	50	55	55	70	80	80
15	10	0.1	150	150	150	150	150	150	150	150	150	150	150	150
16	11	0.05	20	20	20	20	20	20	20	20	20	20	20	20

## StandardSoilParaReader

Eine Komponente zum Auslesen von Bodeninformation aus einem ASCII-File.

### Metainformationen

Klassifikation	
Paket	org.unijena.j2k.io
Autor	Sven Kralisch
Version	1.0_0
Datum	2005-11-10

## Attribute

### Eingabe

Name	Beschreibung	Einheit	Typ
stFileName	Bodentypenparamter Dateiname		String
hrus	Liste der HRU-Objekte		EntityCollection

## Beschreibung der Komponente

Diese Komponente liest Bodeninformationen aus einer ASCII-Datei, die durch *stFileName* identifiziert wird. Für jede Einheit der Liste der *hrus* wählt sie einen Bodentyp, der durch sein Attribut *soilID* identifiziert wird und ergänzt die verbundenen Bodeninformationen zu dem Attributeset der Einheit. Ein Beispiel für eine hydrogeologische Parameterdatei wird in folgender Abbildung (Abbildung 10) gezeigt.

**Abbildung 10. Ausschnitt einer ASCII-Beispieldatei mit Bodeninformationen**

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	# soil parameters for the Gera catchment, derived by Manfred Fink 10/2004, created 27-02-06																
2	SID	depth	kf_min	depth_min	kf_max	cap_rise	aircap	fc_sum	fc_1	fc_2	fc_3	fc_4	fc_5	fc_6	fc_7	fc_8	fc_9
3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	-9999	-9999	-9999	-9999	-9999	1	-9999	-9999	-9999	-9999	-9999	-9999	-9999	-9999	-9999	-9999	-9999
5	n/a	cm	m/d	m	m/d	0/1	mm	mm	mm/dm	mm/dm	mm/dm	mm/dm	mm/dm	mm/dm	mm/dm	mm/dm	mm/dm
6	11	135	0.32	0.5	0.75	0	128.4	77.9	16.4	10.2	10.2	6.9	3.6	3.6	3.6	3.6	3.6
7	12	100	0.2	0.1	0.2	0	25.6	50	21.2	3.2	3.2	3.2	3.2	3.2	3.2	3.2	3.2
8	21	140	0.32	0.5	0.75	0	77.4	90.8	17.4	17.4	6	6	6	6	6	2.6	2.6
9	22	160	0.21	0.5	0.43	0	48.7	91.7	20.2	8.5	8.5	8.5	8.5	8.5	2.9	2.9	2.9
10	23	100	0.2	0.2	0.2	0	32.8	63.4	18.9	18.9	3.2	3.2	3.2	3.2	3.2	3.2	3.2
11	31	180	0.21	0.5	0.43	0	74.1	132.1	20.2	20.2	20.2	8.5	8.5	8.5	8.5	8.5	2.9
12	32	100	0.2	0.3	0.2	0	44.8	86	21.2	21.2	21.2	3.2	3.2	3.2	3.2	3.2	3.2
13	41	160	0.32	0.5	0.75	0	164.9	161.4	22.6	22.6	22.6	19.2	19.2	19.2	3.6	3.6	3.6
14	42	190	0.14	0.5	0.2	0	66.4	141.9	21.2	21.2	8	8	8	8	8	8	8
15	43	100	0.2	0.4	0.2	0	49.6	94.8	18.9	18.9	18.9	18.9	3.2	3.2	3.2	3.2	3.2
16	44	100	0.1	0.2	0.1	0	17.8	62.6	19.7	19.7	2.9	2.9	2.9	2.9	2.9	2.9	2.9
17	51	110	0.1	0.7	0.31	0	46.9	113.6	23.5	23.5	23.5	23.5	2.8	2.8	2.8	2.8	2.8
18	52	180	0.75	0.3	2.29	0	76.5	174.5	24	24	24	34.2	34.2	34.2	0	0	0
19	53	100	0.1	0.3	0.1	0	23.2	79.4	19.7	19.7	19.7	2.9	2.9	2.9	2.9	2.9	2.9
20	54	140	0.08	0.5	0.41	0	51.6	132.3	18	18	18	9.3	6.9	6.9	6.9	6.9	6.9
21	55	130	0.14	0.5	0.2	0	66.1	132.1	21.2	21.2	21.2	11.2	11.2	11.2	11.2	11.2	2.5
22	61	150	0.14	0.5	0.2	0	78.1	154.9	21.2	21.2	21.2	21.2	9.6	9.6	9.6	9.6	9.6
23	62	190	0.21	0.5	0.43	0	108.8	189.8	20.2	20.2	20.2	20.2	16	16	16	16	16
24	63	110	0.03	0.5	0.1	0	22.6	85.2	19.7	19.7	8.7	8.7	8.7	8.7	2.2	2.2	2.2
25	64	180	0.32	0.5	0.75	0	626.5	354.5	24	24	24	34.2	34.2	34.2	18	18	18
26	65	190	0.21	0.5	0.43	0	124	201	23	23	23	23	16	16	16	16	16
27	71	180	0.14	0.5	0.2	0	121	272.5	22.5	22.5	22.5	26.7	26.7	26.7	12.5	12.5	12.5

## SubbasinFlooding

### Metainformationen

Klassifikation	
Paket	flooding
Autor	Sven Kralisch und Markus Meinhardt
Version	1.0_0

Datum	2015-12-11
-------	------------

## Attribute

### Eingabe

Name	Beschreibung	Einheit	Typ
subbasinHRUs	The HRUs that drain into the reach		EntityCollection
elevationAttributeName	The name of the HRU's elevation attribute	m	String
areaAttributeName	The name of the HRU's elevation attribute	sq m	String
floodVolumeAttributeName	The name of the HRU's flood volume attribute that should be used for flooding	liter	String
reachWidth	reach width attribute	m	Double
reachLength	reach length attribute	m	Double
reachHeight	reach height attribute	m	Double
deepening	reach deepening attribute	m	Double

### Status

Name	Beschreibung	Einheit	Typ
runoffComponents	Runoff components of the reach that should be used for flooding	liter	Double[]
floodHeight	The name of the HRU's flood height attribute that was used for flooding	cubic m	Double

## SumAggregator

Eine Komponente zur Berechnung von Summen über numerischen Daten.

### Metainformationen

Klassifikation	
Paket	org.unijena.j2k.aggregate
Autor	Sven Kralisch
Version	1.0_0
Datum	2005-05-22

## Attribute

### Eingabe

Name	Beschreibung	Einheit	Typ
value	Liste der Eingabewerte		Double[]
weight	null		Double

## Status

Name	Beschreibung	Einheit	Typ
sum	Liste der Summenwerte		Double[]

## Beschreibung der Komponente

Diese sehr einfache Komponente dient zur Berechnung von Summen über Attribute des Typs `Attribute.Double`. Um diese Berechnung gleich für mehrere Werte durchführen zu können, nimmt die Komponente mit dem Attribut *value* ein Array von `Attribute.Double`-Werten auf und stellt die Summenwerte entsprechend in einem Array von `Attribute.Double`-Werten (Attribut *sum*) zur Verfügung. Bei jedem Aufruf der Komponente wird dabei das *i*-te Element von *value* zum *i*-ten Element von *sum* hinzuaddiert. Diese Komponente kann z.B. eingesetzt werden, um die Gesamtfläche eines hydrologischen Einzugsgebietes als Summe der Einzelflächen aller räumlichen Modelleinheiten zu berechnen.

## TSDataStoreReader

Eine Komponente zum Auslesen von Zeitreihendaten aus einem JAMS-Datenspeicher.

## Metainformationen

Klassifikation	
Paket	jams.components.io
Autor	Sven Kralisch
Version	1.2
Datum	2014-02-16

## Attribute

### Eingabe

Name	Beschreibung	Einheit	Typ
id	ID des Datenspeichers		String
timeInterval	Zeitintervall, in dem die Komponente Daten aus dem Datenspeicher lesen soll		TimeInterval
calcAvg	null		Boolean
time	null		Calendar

### Ausgabe

Name	Beschreibung	Einheit	Typ
dataSetName	Name des Datensatzes (gleichzusetzen mit ID des Datenspeichers)		String

Name	Beschreibung	Einheit	Typ
dataArray	Array von Doppelwerten, die aus dem Datenspeicher stammen; Reihenfolge je nach Definition des Datenspeichers		DoubleArray
name	null		StringArray
columnID	null		StringArray
elevation	Array von Höhenlagen der Stationen		DoubleArray
xCoord	Array von x-Koordinaten der Stationen		DoubleArray
yCoord	Array von y-Koordinaten der Stationen		DoubleArray
regCoeff	Regressionskoeffizienten		DoubleArray

## Beschreibung der Komponente

Dies Komponente kann zum Auslesen numerischer Zeitreihendaten aus einem Datenspeicher von Zeitreihen-Eingabedaten benutzt werden, der zeitliche Daten aus den Messstationen bereitstellt. Der Datenspeicher wird durch einen ID-Wert identifiziert (*id*), der der Komponente bereitgestellt wird.

In der Anfangsphase wird die Komponente zuerst überprüfen, ob die Daten des Datenspeichers sich mit dem gewünschten Zeitintervall, definiert durch *timeInterval*, überschneiden. Dabei überspringt es die vorderen Datensätze, bis es das Datum erreicht, das als Startdatum des Intervalls definiert wurde. Dann wird die Komponente überprüfen, ob die gewünschte zeitliche Auflösung, definiert durch *timeInterval*, mit dem Datenspeicher übereinstimmt. Letztlich liest die Komponente verschiedene Metadaten aus (z.B: x-/y-Koordinaten und Höhenlagen der Stationen) und stellt sie über die Arrays *xCoord*, *yCoord* und *elevation* bereit.

Während der Verlaufsphase liest die Komponente die Datensätze des Datenspeichers, jeweils einen nach dem anderen, und stellt die Daten durch das Attribut *dataArray* bereit. Abhängig von dem Wert des Attributs *dataCaching* wird die Komponente die linearen Regressionskoeffizienten zwischen den Höhenlagen der Stationen und den aktuellen Datenwerten errechnen. Die Regressionskoeffizienten werden als ein Doppel-Array aus drei Elementen bereitgestellt, was

1. die Steigung der linearen Regressionsfunktion im ersten Element,
2. die Interzeption des linearen Regressionsfunktion mit der x-Achse im zweiten Element und
3. das Bestimmtheitsmaß ( $r^2$ ) im letzten Element

beinhaltet. Die Regressionskoeffizienten werden abhängig von den folgenden möglichen Werten des Attributs *dataCaching* berechnet:

- 0: die Regressionskoeffizienten werden berechnet und zu einer cache-Datei geschrieben im Verzeichnis *tmp* des Arbeitsbereichs des Modells,
- 1: die Regressionskoeffizienten werden nicht berechnet, sondern aus der cache-Datei im Verzeichnis *tmp* des Arbeitsbereichs des Modells ausgelesen,
- 2: die Regressionskoeffizienten werden berechnet und keine cache-Datei wird geschrieben.

## TSPlot

Eine Komponente zur grafischen Darstellung zeitvarianter Daten.

## Metainformationen

Klassifikation	
Paket	jams.components.gui
Autor	Sven Kralisch
Version	1.0_2
Datum	2013-04-26

## Attribute

### Eingabe

Name	Beschreibung	Einheit	Typ
plotTitle	Diagrammtitel		String
titleLeft	Legendentitel der Graphen auf der linken y-Achse (statischer Teil)		StringArray
varTitleLeft	Legendentitel der Graphen auf der linken y-Achse (dynamischer Teil, wird an den statischen Teil (titleLeft) angehängt)		StringArray
titleRight	Statischer Legendentitel der Graphen auf der rechten y-Achse		StringArray
colorLeft	Farben der Graphen auf der linken Achse, gültige Werte siehe Beschreibung		StringArray
colorRight	Farben der Graphen auf der rechten Achse, gültige Werte siehe Beschreibung		StringArray
typeLeft	Typ der Graphen auf der linken Achse, Codierung siehe Beschreibung		Integer
typeRight	Typ der Graphen auf der rechten Achse, Codierung siehe Beschreibung		Integer
xAxisTitle	Titel der x-Achse		String
leftAxisTitle	Beschriftung der linken y-Achse		String
rightAxisTitle	Beschriftung der rechten y-Achse		String
rightAxisInverted	Flag zur Invertierung der rechten y-Achse		Boolean
dateFormat	Formatstring zur Definition der Darstellung des Datums (z.B. dd-MM-yyyy - siehe JavaDoc von SimpleDateFormat)		String
time	aktuelle Zeit		Calendar
valueLeft	Werte zur Darstellung auf der linken y-Achse		Double[]
valueRight	Werte zur Darstellung auf der rechten y-Achse		Double[]
cacheSize	Größe des Caches, vor der nächsten Aktualisierung des Diagramms werden bis zu cacheSize Werte zwischengespeichert, um die Runtime-Performance zu steigern		Integer

Name	Beschreibung	Einheit	Typ
paintGridLines	null		Boolean
legendRight	null		Boolean

## Beschreibung der Komponente

### Konfiguration

Diese Komponente dient zur grafischen Darstellung zeitvarianter Daten. Dabei kann pro Zeitschritt eine beliebige Anzahl von Fließkomma-Daten auf der linken und rechten y-Achse grafisch ausgegeben werden. Auf der x-Achse wird die aktuelle Zeit dargestellt. Alle Achsentypen und -beschriftungen können dabei konfiguriert werden, ebenso die Farben und Typen der dargestellten Graphen. Folgende Farbwerte sind dabei zulässig: *yellow, orange, red, pink, magenta, cyan, yellow, green, lightgray, gray, black*.

Die Graphentypen können getrennt für die linke und rechte y-Achse angegeben werden (*typeLeft/typeRight*). Nachfolgende Tabelle listet die möglichen Typen auf.

**Tabelle 1. Erlaubte Graphentypen**

Codierung	Diagrammtyp
0	XYLineAndShapeRenderer
1	XYBarRenderer
2	XYAreaRenderer
3	XYLineAndShapeRenderer
4	XYDotRenderer
5	XYDifferenceRenderer ( <b>Achtung:</b> bei Verwendung dieses Typs müssen mindestens zwei Graphen auf der entsprechenden Achse dargestellt werden!)
6	XYStepRenderer
7	XYStepAreaRenderer

Da es sich bei *valueLeft/valueRight* um Arrays handelt, kann theoretisch eine beliebige Anzahl von Graphen dargestellt werden. In der Praxis wird jedoch die Laufzeitperformance mit steigender Anzahl von Graphen negativ beeinflusst. Weiterhin ist zu berücksichtigen, dass die Anzahl der Farbwerte (*colorLeft/colorRight*), die Anzahl der Bezeichnungen der einzelnen Graphen (*titleLeft/titleRight*) und die Anzahl der darzustellenden Werte (*valueLeft/valueRight*) jeweils übereinstimmen. So führt beispielsweise die Angabe nur eines rechten Farbwertes bzw. nur eines rechten Titels zum Fehler, wenn zwei rechte Graphen existieren.

### Verhalten

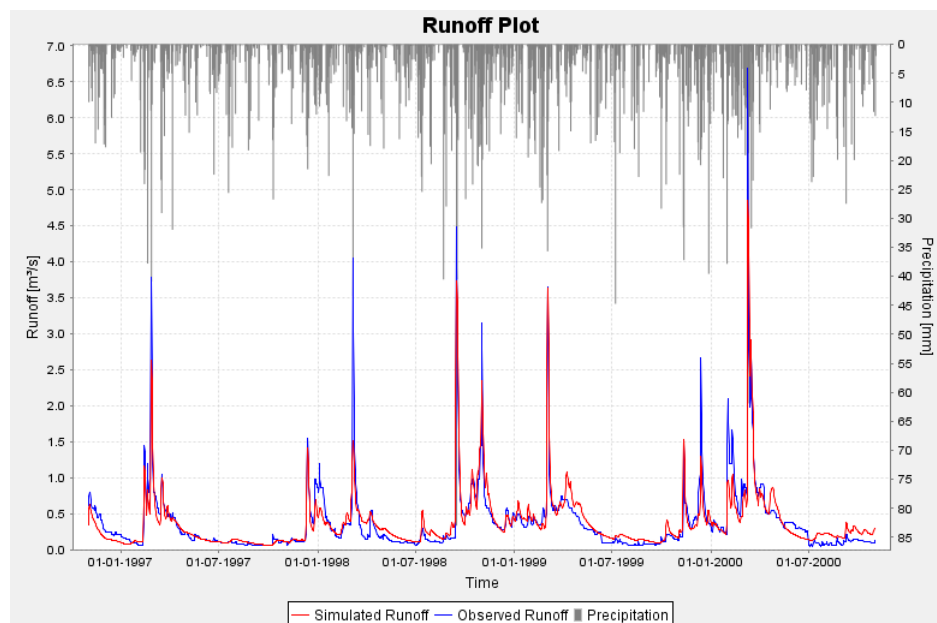
Bei jedem Aufruf der *run()*-Methode werden die an TSPlot übergebenen Daten (*valueLeft/valueRight*) dynamisch in das Diagramm eingefügt und entsprechend des definierten Typs (*typeLeft/typeRight*) und der definierten Farbe (*colorLeft/colorRight*) graphisch ausgegeben. Zu jedem Zeitpunkt kann der Nutzer dabei mit der TSPlot-Komponente interagieren, folgende Aktionen sind möglich:

- **Hereinzoomen:** Hierzu wird an der oberen linken Ecke des gewünschten Zoombereiches mit der linken oder rechten Mousetaste in das Diagrammfenster geklickt, die Mouse bei gehaltener Taste nach unten rechts gezogen und die Taste am gewünschten Endpunkt wieder losgelassen.

- **Herauszoomen:** Hierzu wird ähnlich wie beim Hereinzoomen vorgegangen, d.h. die Mouse wird bei gedrückter Taste von unten rechts nach oben links gezogen und losgelassen.
- **Aktuelle Ansicht in Datei speichern:** Durch einfachen Mouseklick mit der rechten Taste in das Diagrammfenster wird das TSPlot-Menü geöffnet, dort der Menüpunkt "Speichern unter..." gewählt und eine Ausgabedatei festgelegt. Die Ausgabe erfolgt im PNG-Format für den aktuell sichtbaren Bereich des Diagramms.
- **Darstellung anpassen:** Durch einfachen Mouseklick mit der rechten Taste in das Diagrammfenster wird das TSPlot-Menü geöffnet und dort der Menüpunkt "Eigenschaften..." gewählt. Im darauffolgenden Dialogfenster können verschiedene Darstellungsparameter des Diagramms angepasst werden.

Abbildung 11 zeigt das Beispiel einer TSPlot-Komponente mit zwei linken und einem rechten Graphen. Der rechte Graph wurde dabei invertiert (*rightAxisInverted*) dargestellt.

**Abbildung 11. Beispiel einer TSPlot-Komponente mit zwei linken und einem rechten Graphen**



## TemporalSumAggregator

Eine Komponente für die gewichtete Aggregation von zeitvarianten Daten in einem Zeitintervall.

## Metainformationen

Klassifikation	
Paket	jams.components.aggregate
Autor	Sven Kralisch
Version	1.0_0
Datum	2006-07-19



## Attribute

### Eingabe

Name	Beschreibung	Einheit	Typ
time	aktuelle Zeit		Calendar
value	Werte, die aggregiert werden sollen		Double[]
weight	ein Gewicht zur Berechnung des gewichteten Aggregatwerts		Double
aggregationTimeInterval	ein Zeitintervall, der Beginn und Ende der gewichteten zeitlichen Aggregation definiert		TimeInterval[]
months	null		Integer[]
average	Berechnung des Durchschnittswerts. Falls dieser Wert falsch ist, wird die Summe berechnet		Boolean

### Status

Name	Beschreibung	Einheit	Typ
sum	der resultierende gewichtete Aggregatwert		Double[]

## Beschreibung der Komponente

Diese Komponente bietet eine Aggregationsfunktion für eine oder mehr zeitvariante Datenwerte. Die Komponente nimmt den aktuellen Zeitschritt ( *time* ), einen Zeitintervall ( *aggregationTimeInterval* ), den/die zu aggregierenden Wert(e) ( *value* ), einen Gewichts-Wert ( *weight* ) und das aktuelle berechnete Aggregat ( *aggregate* ) als Eingabewerte und berechnet ein neues Aggregat, wenn der aktuelle Zeitschritt im Zeitintervall ist. Wenn der Parameter *average* auf wahr gesetzt ist (default), wird das Aggregat die (gewichteten) Durchschnittswerte der Eingabewerte halten oder die (gewichtete(n)) Summe(n).

Ein typischer Anwendungsfall ist die Berechnung von Durchschnittswerten für simulierte Werte von räumlichen Modelleinheiten (z.B. die berechnete Verdunstung in mm) innerhalb einer bestimmten Zeitspanne, z.B. ein Jahr. Das Ergebnis wäre in diesem Fall ein Durchschnittswert pro HRU, ein typischer Wert für das Gewicht ( *weight* ) wäre die HRU-Fläche.

## VariableAdder

### Metainformationen

Klassifikation	
Paket	org.unijena.j2k
Autor	Peter Krause
Version	1.0_0
Datum	

## Attribute

### Eingabe

Name	Beschreibung	Einheit	Typ
inVars	the input variables to add		Double[]

### Ausgabe

Name	Beschreibung	Einheit	Typ
outVar	the result of the summation		Double

## WeightedSumAggregator

Eine Komponente zur Berechnung von gewichteten Summen über numerischen Daten.

### Metainformationen

Klassifikation	processing.aggregate
Paket	org.unijena.j2k.aggregate
Autor	Sven Kralisch
Version	1.0_0
Datum	

## Attribute

### Eingabe

Name	Beschreibung	Einheit	Typ
value	Liste der Eingabewerte		Double[]
weight	ein Gewicht zur Berechnung der gewichteten Summen		Double

### Status

Name	Beschreibung	Einheit	Typ
sum	Liste der Summenwerte		Double[]

## Beschreibung der Komponente

Diese sehr einfache Komponente dient zur Berechnung von gewichteten Summen über Attribute des Typs `Attribute.Double`. Um diese Berechnung gleich für mehrere Werte durchführen zu können, nimmt die Komponente mit dem Attribut *value* ein Array von `Attribute.Double`-Werten auf und stellt die Summenwerte entsprechend in einem Array von `Attribute.Double`-Werten (Attribut *sum*) zur Verfügung. Bei jedem Aufruf der Komponente wird dabei das *i*-te Element von *value* mit *weight* multipliziert und das Ergebnis zum *i*-ten Element von *sum* hinzuaddiert.

# Modellparameter

<b>AhumDataReader</b>	
id	rhum
<b>AhumRegionaliser</b>	
fixedMinimum	0
nidw	3
rsqThreshold	0.75
elevationCorrection	false
<b>CalcDailySolarRadiation</b>	
angstrom_b	0.50
angstrom_a	0.25
tempRes	d
<b>CalcLatLong</b>	
projection	UTM34L
<b>CalcPotentialET</b>	
tempRes	d
<b>CatchmentResetter</b>	
value	0
<b>DoubleConditionalContext</b>	
value2	0
<b>ETPlot</b>	
colorLeft	red;blue:green
dateFormat	dd-MM-yyyy hh:mm
xAxisTitle	Time
typeLeft	0
titleLeft	actET;potET;refET
leftAxisTitle	ET [mm]
<b>Efficiencies</b>	
effMethod	1,2,3,4,5,6,7,8,9,10,11,12
<b>EntityReader</b>	
reachFileName	parameter/reach.par
hruFileName	parameter/hrus.par
upStreamCatchmentAttribute	upstreamCatchment
<b>ExtRad</b>	
longTZ	15
locGrw	e
tempRes	d
<b>GWReader</b>	

gwFileName	parameter/hgeo.par
<b>GeometryReader</b>	
idName	HRUID
shapeFileName	input/local/gis/hrus/hrus.shp
<b>HRU2HRURouting</b>	
inNames	inRD1;inRD2;inRG1;inRG2
<b>HRU2ReachRouting</b>	
inNames	inRD1;inRD2;inRG1;inRG2
<b>HRUResetter</b>	
value	0
<b>InitGroundWater</b>	
initRG1	0.040008
initRG2	0.025265
<b>InitSoilWater</b>	
FCAdaptation	1.284269
ACAdaptation	1.295612
<b>J2K</b>	
snow_trs	-0.7
timeInterval	1958-10-01 07:30 2013-10-01 07:30 6 1
snow_trans	1.34
<b>J2KProcessGroundwater</b>	
gwRG2Fact	3.172203
gwCapRise	0.269081
gwRG1RG2dist	0.639081
gwRG1Fact	1.968105
<b>J2KProcessInterception</b>	
a_rain	0.200443
a_snow	2.78
<b>J2KProcessLumpedSoilWater</b>	
soilMaxPerc	5.595396
soilLatVertLPS	0.836812
soilConcRD2	9.589821
soilLinRed	0.956092
soilDistMPSLPS	0.495699
soilImpLT80	0.01541
soilMaxDPS	2.87151
soilPolRed	5
soilMaxInfSnow	40.0
soilImpGT80	0.191269

soilDiffMPSLPS	0.002203
soilConcRD1	4.034305
soilMaxInfSummer	100.729498
soilOutLPS	1.935329
soilMaxInfWinter	155.128085
<b>J2KProcessLumpedSoilWater_flood</b>	
soilMaxPerc	1.355756
soilLatVertLPS	4.153037
soilConcRD2	8.698757
soilLinRed	0.483038
soilDistMPSLPS	5.717994
soilImpLT80	0.01541
soilMaxDPS	201.886932
soilPolRed	5.0
soilMaxInfSnow	40.0
soilImpGT80	0.191269
soilDiffMPSLPS	0.107243
soilConcRD1	2.694925
soilMaxInfSummer	166.429397
soilOutLPS	2.876461
soilMaxInfWinter	43.974482
<b>J2KProcessReachRouting</b>	
slopeAsProportion	true
tempRes	d
flowRouteTA	2.552779
<b>J2KProcessSnow</b>	
baseTemp	0.3
active	false
ccf_factor	0.0012
r_factor	0.210
t_factor	0.840
g_factor	1.739
snowCritDens	0.381
<b>LUReader</b>	
luFileName	parameter/landuse.par
<b>Map</b>	
shapeFileName1	input/local/gis/streams/streams.shp
stylesFileName	input/local/gis/styles.sld
rangeColor	red;red;red;red;red;red;red;red;red;red;red;red

styleID	0
numOfRanges	10;10;10;10;10;10;10;10;10;10;10
showAttr	outRD1_avg;outRD2_avg;outRG1_avg;outRG2_avg;actET_avg;potE
shapeColors	0000FF
<b>Map3D</b>	
stylesFileName	input/local/gis/styles.sld
rangeColor	green;red;blue
showAttr	outRD1_avg;outRD2_avg;outRG1_avg
shapeColors	0000FF
resolution	256
shapeFileName1	input/local/gis/streams/streams.shp
heightMap	input/local/gis/dgm_gera.asc
light	true
styleID	0
numOfRanges	10;10;10
<b>ObsRunoff</b>	
arrayIndex	0
<b>ObsRunoffDataReader</b>	
id	orun
<b>Percolation</b>	
colorLeft	red;blue;green
dateFormat	dd-MM-yyyy hh:mm
xAxisTitle	Time
typeLeft	0
titleLeft	actRG1;actRG2;percolation
leftAxisTitle	GW [mm]
<b>PrecipDataReader</b>	
id	rain
<b>PrecipRegionaliser</b>	
fixedMinimum	0
nidw	5
rsqThreshold	0.75
elevationCorrection	false
<b>RCRichter</b>	
tempNIDW	1
regThres	0.75
pIDW	2.0
<b>RG2</b>	
colorLeft	red;blue;green

dateFormat	dd-MM-yyyy hh:mm
xAxisTitle	Time
typeLeft	0
titleLeft	catchmentRG2_w;outRG2
leftAxisTitle	Groundwater [mm]
<b>Radiation</b>	
colorLeft	red;blue;green
dateFormat	dd-MM-yyyy hh:mm
xAxisTitle	Time
typeLeft	0
titleLeft	netRad;solRad;sunh
leftAxisTitle	Rad
<b>RaincorrDataReader</b>	
id	raincorr
<b>RaincorrRegionaliser</b>	
fixedMinimum	0
nidw	5
rsqThreshold	0.75
elevationCorrection	false
<b>Rhum</b>	
colorLeft	blue
dateFormat	dd-MM-yyyy hh:mm
xAxisTitle	Time
typeLeft	0
titleLeft	rhum
leftAxisTitle	rel. humidity [%]
<b>Runoff (components)</b>	
colorLeft	red;blue;green;black
dateFormat	dd-MM-yyyy
xAxisTitle	Time
typeLeft	0
rightAxisTitle	Precipitation [mm]
titleLeft	Overlandflow;Interflow;Groundwater (fast);Groundwater (slow)
titleRight	Precipitation
typeRight	1
colorRight	gray
rightAxisInverted	1
leftAxisTitle	Hydrological Component [mm]

<b>Runoff (overall)</b>	
colorLeft	red;blue
dateFormat	dd-MM-yyyy
xAxisTitle	Time
typeLeft	0
rightAxisTitle	Precipitation [mm]
titleLeft	Simulated Runoff;Observed Runoff
titleRight	Precipitation
typeRight	1
colorRight	gray
rightAxisInverted	1
leftAxisTitle	Runoff [m <sup>3</sup> /s]
<b>STReader</b>	
stFileName	parameter/soils.par
<b>SimRunoffConverter</b>	
outUnit	m <sup>3</sup> /s
inUnit	L/day
<b>Snow</b>	
colorLeft	blue
dateFormat	dd-MM-yyyy hh:mm
xAxisTitle	Time
typeLeft	0
titleLeft	snowTotSWE
leftAxisTitle	snow water equivalent
<b>Soil moisture act</b>	
colorLeft	red;blue;green
dateFormat	dd-MM-yyyy hh:mm
xAxisTitle	Time
typeLeft	0
titleLeft	actLPS;actMPS
leftAxisTitle	act. saturation
<b>Soil moisture rel</b>	
colorLeft	red;blue
dateFormat	dd-MM-yyyy hh:mm
xAxisTitle	Time
typeLeft	0
titleLeft	satMPS;satLPS
leftAxisTitle	rel. saturation
<b>Storages</b>	



colorLeft	red;blue;green;black
dateFormat	dd-MM-yyyy hh:mm
xAxisTitle	Time
typeLeft	0
titleLeft	actDPS;actDPS2;channelstorage;floodVolume
<b>SunhDataReader</b>	
id	sunh
<b>SunhRegionaliser</b>	
fixedMinimum	0
nidw	3
rsqThreshold	0
elevationCorrection	false
<b>Temp</b>	
colorLeft	red;blue;green
dateFormat	dd-MM-yyyy hh:mm
xAxisTitle	Time
typeLeft	0
titleLeft	Tmax;Tmean;Tmin
leftAxisTitle	Temp [°C]
<b>TemporalSumAggregator1</b>	
average	true
<b>TmaxDataReader</b>	
id	tmax
<b>TmaxRegionaliser</b>	
fixedMinimum	-273
nidw	3
rsqThreshold	0.75
elevationCorrection	false
<b>TmeanDataReader</b>	
id	tmean
<b>TmeanRegionaliser</b>	
fixedMinimum	-273
nidw	3
rsqThreshold	0.75
elevationCorrection	false
<b>TminDataReader</b>	
id	tmin
<b>TminRegionaliser</b>	
fixedMinimum	-273

nidw	3
rsqThreshold	0.75
elevationCorrection	false
<b>Wind</b>	
colorLeft	blue
dateFormat	dd-MM-yyyy hh:mm
xAxisTitle	Time
typeLeft	0
titleLeft	Wind
leftAxisTitle	Wind [m/s]
<b>WindDataReader</b>	
id	wind
<b>WindRegionaliser</b>	
fixedMinimum	0
nidw	3
rsqThreshold	0.75
elevationCorrection	false
<b>ZeroSetter</b>	
value	0
<b>ZeroSetter_1</b>	
value	0
<b>ahumWeights</b>	
dataStoreID	rhum
pidw	2.0
<b>j2kArrayGrabber</b>	
tempRes	d
<b>precipWeights</b>	
dataStoreID	rain
pidw	2.0
<b>raincorrWeights</b>	
dataStoreID	raincorr
pidw	2.0
<b>sunhWeights</b>	
dataStoreID	sunh
pidw	2.0
<b>tmaxWeights</b>	
dataStoreID	tmax
pidw	2.0
<b>tmeanWeights</b>	

dataStoreID	tmean
pidw	2.0
<b>tminWeights</b>	
dataStoreID	tmin
pidw	2.0
<b>windWeights</b>	
dataStoreID	wind
pidw	2.0

# Literaturverzeichnis

- [Allen et.al 1998] R.G. Allen, L. Pereira, D. Raes und M. Smith. *Crop evapotranspiration: Guidelines for computing crop water requirements-FAO Irrigation and drainage paper 56*. FAO. Rom. 300. 1998.
- [Bertle 1966] F.A. Bertle. *Effects of Snow Compaction on Runoff from Rain and Snow*. Engineering Monograph. 35. 1-45. 1966.
- [Herrmann 1976] A. Herrmann. *Einfluss des Alpensüdföhns auf die Schneedeckenentwicklung und das nival gesteuerte Abflussgeschehen*. Polarforschung. 46(2). 83-94. 1976.
- [Knauf 1980] D. Knauf. *Die Berechnung des Abflusses aus einer Schneedecke. Analyse und Berechnung oberirdischer Abflüsse*. DVWK-Schriften. 46. 83-94. 1980.
- [Krause 2001] P. Krause. *Das hydrologische Modellsystem J2000: Beschreibung und Anwendung in großen Flußeinzugsgebieten*. Schriften des Forschungszentrum Jülich: Reihe Umwelt/Environment. Band 29. 2001.
- [Krause et al. 2005] P. Krause, D. Boyle und F. Bäse. *Comparison of different efficiency criteria for hydrological model assessment*. Advances in Geosciences. 5. 89-97. 2005.
- [Kuchment et al. 1983] L.S. Kuchment, V.N. Demidov und Y.G. Motovolov. *Formirovanie rechnogo stok (fiziko-matematicheskde modeli) (River runoff formation/physically based models) (in Russian)*. Nauka. Moscow. 1983.
- [Lang 2005] H. Lang. *Hydrometeorologische Ergebnisse aus Abflussmessungen im Bereich des Hintereisferners (Ötztaler Alpen) in den Jahren 1957 bis 1959*. Archiv für Meteorologie. Serie B, Band 14. 280-302. 2005.
- [Monteith 1975] J.L. Monteith. *Vegetation and atmosphere, Vol.1 Principles*. Academic Press. London. 1975.
- [Nash & Sutcliffe 1970] J. Nash und J. Sutcliffe. *River flow forecasting through conceptual models. Part I: a discussion of principles*. Journal of Hydrology. 10. 282-290. 1970.
- [Richter 1995] D. Richter. *Ergebnisse methodischer Untersuchungen zur Korrektur des systematischen Messfehlers des Hellmann-Niederschlagsmessers*. Deutscher Wetterdienst. 1995.
- [Vehviläinen 1992] B. Vehviläinen. *Snow cover models in operational watershed forecasting. Yhteenvento: Lumimallit vesistöjen ennustemalleissa.. Publications of the Water and Environment Research Institute 11*. National Board of Waters and the Environment. Finland, Helsinki. 1992.
- [Willmot 1981] C.J. Willmot. *On the validation of models*. Physical Geography. 2. 184-194. 1981.
- [kralisch\_coupling\_2009] S. Kralisch, F. Zander und P. Krause. R.S. Anderssen. R.D. Braddock. L.T.H. Newham. *Coupling the RBIS Environmental Information System and the JAMS Modelling Framework*.

„Coupling the RBIS Environmental Information System and the JAMS Modelling Framework“. 902-908. 978-0-9758400-7-8. *Proceedings of the 18th World IMACS Congress and MODSIM09 International Congress on Modelling and Simulation*. <http://www.mssanz.org.au/modsim09/C3/kralisch.pdf>. 2009-07.