

DANUBIA
Software-Documentation

GLOWA-DANUBE Papers

Technical Release No. XXX (Tourism)

Program Name:

DANUBE.Tourist

Sofware-Release-No.:

0.4.0

Documentation:

0.1

Release Date:

July 2004

Responsible Authors:

Alexander Dingeldey

Mario Sax

1. Purpose

1.1 General Purpose

The purpose of DANUBIA.tourism is to supply the tourism water demand based on the relevant tourism infrastructure (swimming pools, golf courses and skiing areas with artificial snow), the tourism suprastructure (accomodation, e.g. hotels, b&bs,...) and the tourisum demand (overnight stays).

1.2 Model Description

The first step of the model is to load all relevant toursim infrastructure and su-prastucture data into the model:

- Swimming pools
- Golf courses
- Skiing areas with artificial snow
- Accomodation (Beds in hotels, B&Bs,...)

The tourism demand consists of two sets of data:

Advanced set of tourism demand data:

Overnigth stays per month based on the real statistical values.

Basic set of tourism demand data:

If there is no advanced data available in a proxel at a time the basic data is used. The yearly tourism demand, represented by the overnigt stays, is calculated on a yearly basis with a regression modell, that has been developed on a time-series analysis. These values are distributed by an individual seasonability function on all months of the year. This basic function will calculate sensible tourism demand data for modell runs for future time periods.

The tourism water demand is calculated on a monthly basis.

2. Input and output data

2.1 Input Values

From the DANUBA Model:

- TourismWaterSupply -> Water supplied for the tourism model

From internal sources / Local files:

data/actor/tourist/proxeltable.txt

Proxel-Based Values:

1	ProxelID	ProxelID			
2	LandID	CountryID			
			1 = BY	4 = T	7 = CH
			2 = BW	5 = S	99 = Div.
			3 = OÖ	6 = VA	
3	KreisID	District ID	Long		
4	GemeindeID	Community ID	Long		
5	BevAntGem	Percentage of the Population of the Community	Double		
6	BevAntKreis	Percentage of the Population of the District	Double		
7	Golf	Number of Golf Course	Int		
8	Schnee	Size of the Area with Artificial Snow	Int		
9	Schwimm	Class of Swimming-Pool		1 = Freibad	
				2 = Frei- und Hallenbad	
				3 = Hallenbad	
				4 = Erlebnisbad	
				5 = Thermalbad	
				6 = Naturbad	

data/actor/tourist/gemeindetable.txt

Row	Variable	Description	Values
1	LandID	CountryID	1 = BY 4 = T 7 = CH 2 = BW 5 = S 99 = Div. 3 = OÖ 6 = VA
2	KreisID	District ID	Long
3	GemeindeID	Community ID	Long
4	Achse	Basis for the Regression Model	Long
5	Wachstum	Growth for the Regression Model	Double
6	Betten	Number of Beds in the Community	Long
7	Betriebe	Number of Hotels/Pensions... in the Community	Long
8	Anteil_Jan	Percentage of Overnightstays in January	Float
9	Anteil_Feb	Percentage of Overnightstays in February	Float
10	Anteil_Mar	Percentage of Overnightstays in March	Float
11	Anteil_Apr	Percentage of Overnightstays in April	Float
12	Anteil_Mai	Percentage of Overnightstays in May	Float
13	Anteil_Jun	Percentage of Overnightstays in June	Float
14	Anteil_Jul	Percentage of Overnightstays in July	Float
15	Anteil_Aug	Percentage of Overnightstays in August	Float
16	Anteil_Sep	Percentage of Overnightstays in September	Float
17	Anteil_Okt	Percentage of Overnightstays in October	Float
18	Anteil_Nov	Percentage of Overnightstays in November	Float
19	Anteil_Dez	Percentage of Overnightstays in December	Float
20	nahmult	Multiplicator for recreation model	Float
21	gastmult	Multiplicator for recreation model – restaurants	Float
22	besmult	Multiplicator for recreation model – visits	Float
23	musmult	Multiplicator for recreation model – museums	Float
24	anschlussgrad	Percentage of supply of drinkingwater System	Float

data/actor/tourist/fein/feindat_****.txt

where **** is the year starting with 1995

Row	Variable	Description	Values
3	GemeindeID	Community ID	Long
8	Ueber_Jan	Overnightstays in January	Long
9	Ueber_Feb	Overnightstays in February	Long
10	Ueber_Mar	Overnightstays in March	Long
11	Ueber_Apr	Overnightstays in April	Long
12	Ueber_Mai	Overnightstays in May	Long
13	Ueber_Jun	Overnightstays in June	Long
14	Ueber_Jul	Overnightstays in July	Long
15	Ueber_Aug	Overnightstays in August	Long
16	Ueber_Sep	Overnightstays in September	Long
17	Ueber_Okt	Overnightstays in October	Long
18	Ueber_Nov	Overnightstays in November	Long
19	Ueber_Dez	Overnightstays in December	Long

2.2 Output Values

To the DANUBIA Model:

- TourismWaterDemand – Water demand of the Tourism Model
- OvernigthStays – Monthly Overnight Stays of the Tourism Model

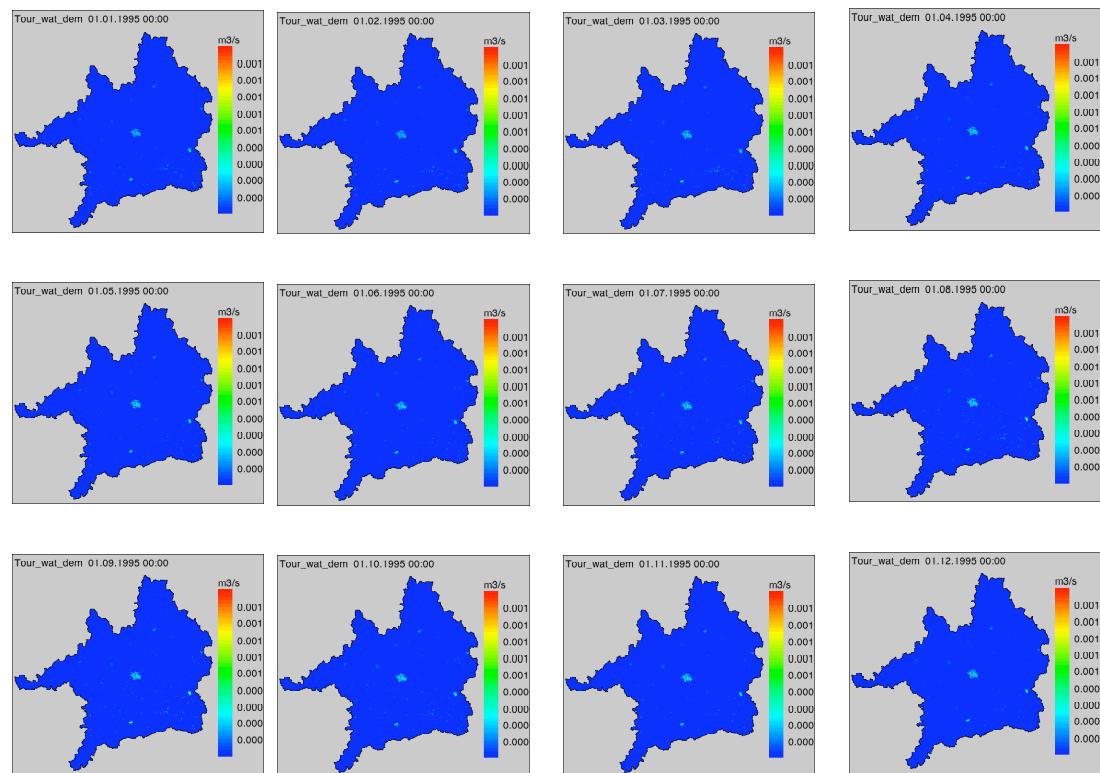
To the Result Interface:

- TourismWaterDemand – Water demand of the Tourism Model

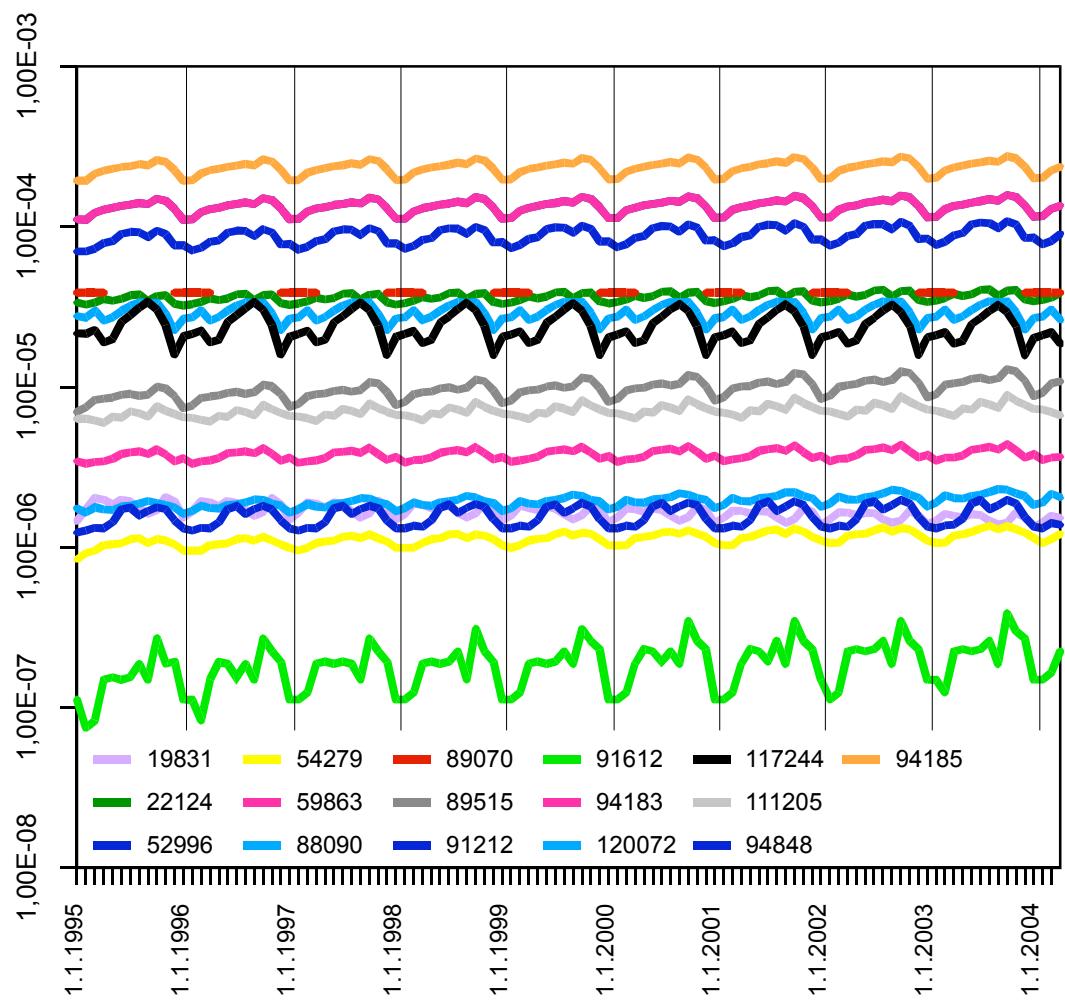
Local Output:

- Monthly ASCII grids of the water demand:
/modelout/text/actour/tourism/*_tourism_demand.asc
- Yearly cumulated yearly water demand:
/modelout/text/actour/tourism/kumjahr_*_.txt
- Monthly graphics of the water demand:
/modelout/jpg/actor/tourism/Tour_wat_dem_*.png
- Time series of the water demand in relevant proxels:
/modelout/timeseries/toursim_*.txt

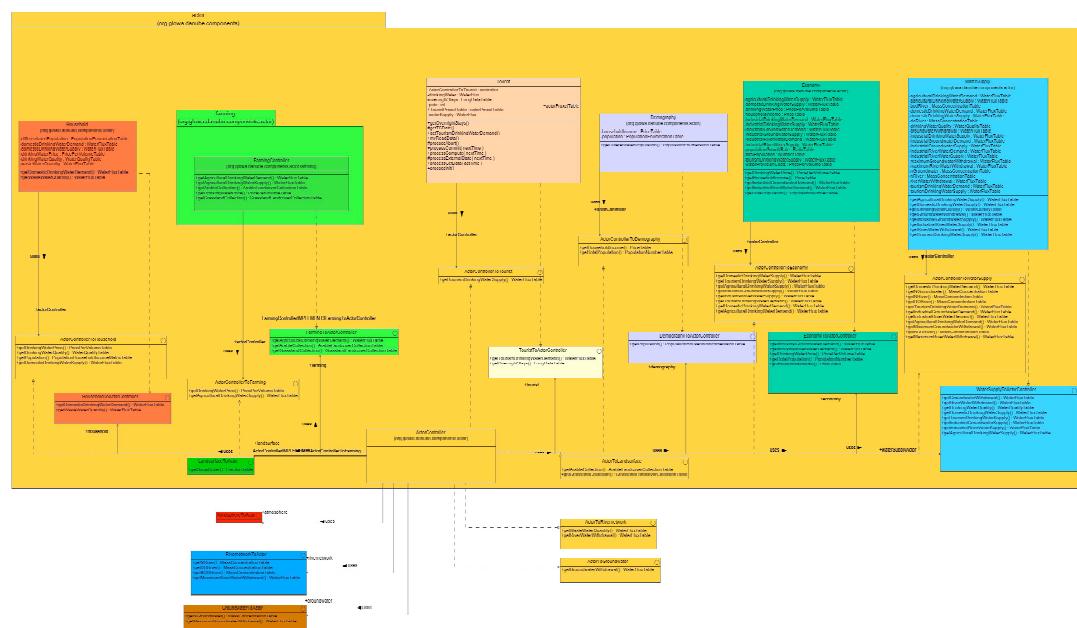
Monthly graphics of the tourism water demand:



Timeseries of tourism water demand in selected proxels:



The Tourism Model in the DANUBIA Context:



3. Technical Description

Init

The first step is the initialisation of all Variables. A data-table with all TouristProxels and Tourist-Communities (TouristGemeinden) is created. The initial values of the basic and the advanced model the tourism infra- and superstructure are read out of the local files and checked to sensibility. The superstructure is distributed to the proxels within the communities according to the population. After loading the initial values the first computation of the water demand is conducted, the results are tested and exported.

GetData

Every January, whether advanced Data is available, it is loaded into the Tourist-Community Table and checked to sensibility.

The next step is the computation of the overnight-stays in every proxel. The overnight stays of the community, either from the advanced model or the basic model are distributed to all Proxels in the community according to the population of within the proxel.

Compute

The tourism water demand is calculated. Every December a file with cumulated values is exported to the filesystem..

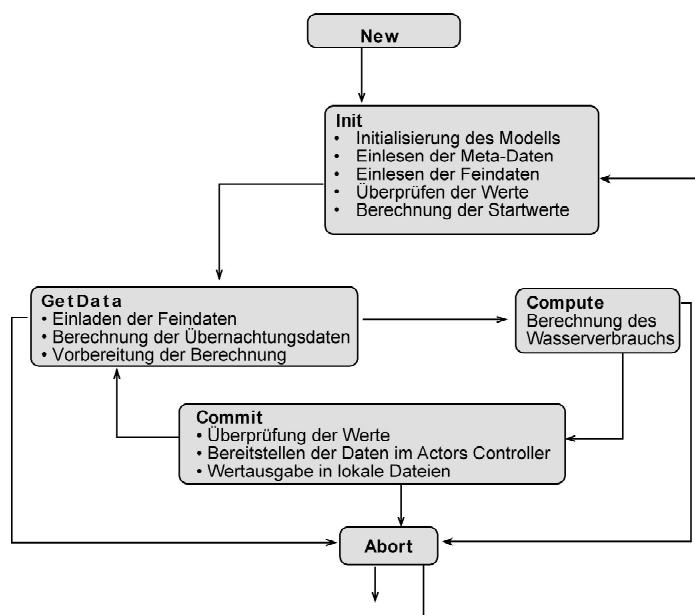
Commit

After the calculations another sensibility test is conducted and the water demand is exported via the TouristsToActorController to the water supply model.

Abort

All variables are set to 0

Model Flow-Chart



4. Components of DANUBIA.Actor.Tourist

File Name:

Tourist.java

The class which controls all the actions taken by the model WaterSupplyActor: The initialisation of all the variables, collection of data, starting of the calculation, making the export data available for the other mode

TouristInfo.java

Provides the meta information about the component

TouristProxel.java

Defines variables and algorithms for the smallest entity of the tourism model.

TouristProxyTable.java

Defines the parameters and properties of a TouristProxelTable

TouristGemeinde.java

Provides information and algorithms for calculations based on a community

TouristGemeindeTable.java

Array of all tourist communities. Provides routines for initialisation and calculation.

TouristResult.java

Defines the result-interface

Utilities/FormatedFilePrinterWS.java Utilities to write ASCII-Grid Output

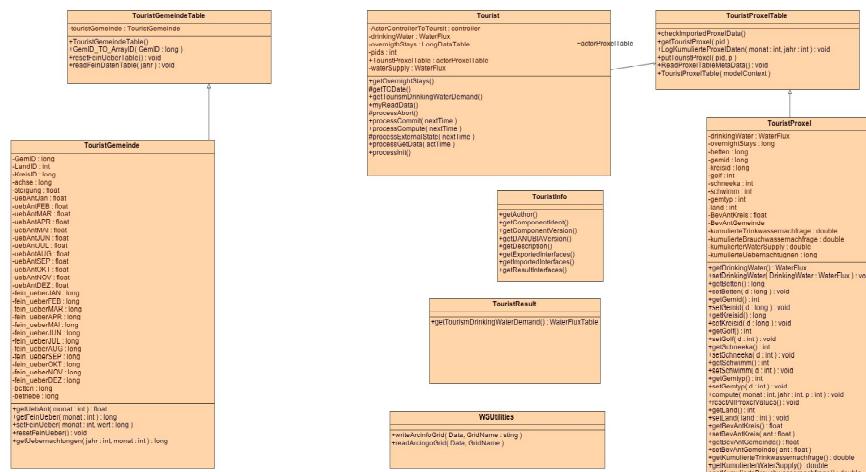
Utilities/FormatedFileReaderWS.java

Utilities.WSUtilities.java

readme.txt

Version History and release notes

The Model-Structure:



5. Compliance with DANUBIA interfaces

The DANUBIA.Actor.Tourist model compiles with the ActorControllerToTourist.java as it was defined in the release 0.92 of the DANUBIA-UML diagram and it is based on the DANUBIA base framework.

The model is programmed, that the probability of a crash caused by errors is reduced to a minimum. For example in case of errors in the file system the model will not crash but set all variables to 0.

All objects of the tourism model are serialisable to implement the DANUBIA recovery functionality.

The Model has been successfully compiled and tested in the DANUBIA Testenvironment.

Javadoc comments are used to describe the code.