**iMoMo hydro-climatological information system**

**What is it?**

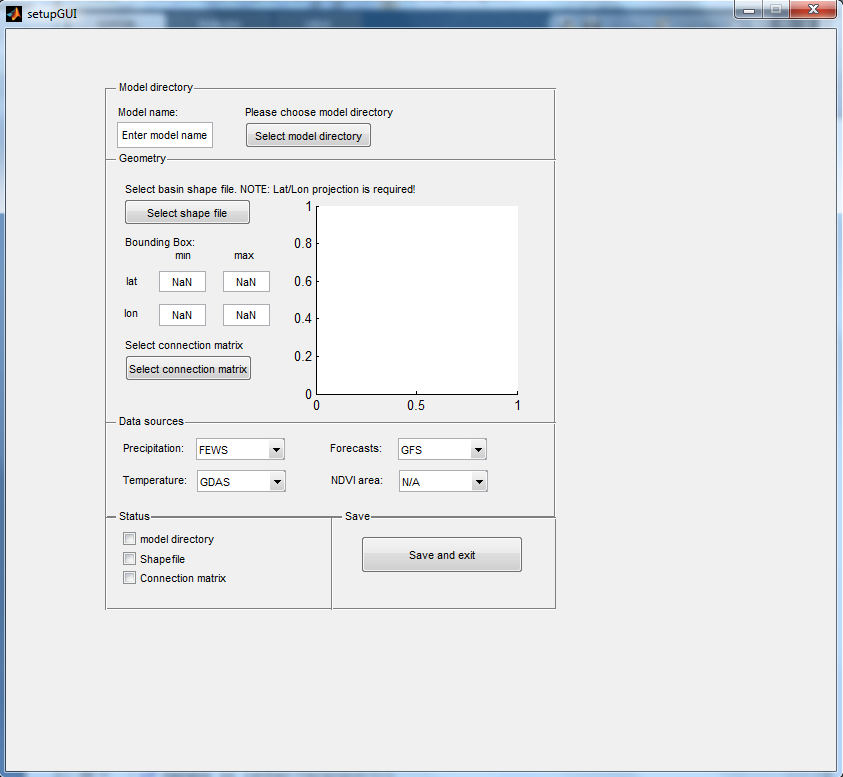
The iMoMo hydro-climatological information system consists of a hydrological model which can simulate water fluxes on a sub-basin scale. Satellite and ground data (meteorological stations and gauging stations data) can be assimilated with the help of the Ensemble-Kalman-Filter to ensure the best possible representation of the current real situation in the catchment. From there, forecast from numerical weather models can be used to estimate the hydrological situation for the near future.

**System requirements**

The software has been developed and tested under Mac OS X. It has not been tested on other operating systems. The software requires a working installation of [MATLAB](http://www.mathworks.com/) (a numerical computing environment) and [openDA](http://www.openda.org/) (an open source data assimilation toolbox), and [python](https://www.python.org/) (a programming language).

**How to set up the system for a new catchment?**

1. **Start the *setupGui.m* file in the folder *src/setup***



*Fig.1. Graphical user interface to set up the system for a new catchment.*

1. **Enter a model name and choose a model directory**

* Enter a model name or a default directory name is chosen.
* Choose the folder RRM/app as model directory
* The folder structure is generated in the chosen directory

1. **Select a 2D basin shape file with a WGS84 coordinate system**

* Check the displayed shape file
* One can manually increase the area of interest by adjusting the bounding box

1. **Select a connection matrix specifying the organization of the subcatchments**

* See for example the connection matrix of the Themi

1. **Select a precipitation data source**

* At the moment only FEWS (see *getFEWS.m*) or TRMM (see *getTRMM.m*) is available

1. **Select a air temperature data source**

* At the moment only GDAS is available (see *getGDAS.m*)

1. **Select a weather forecast data source**

* At the moment only GFS is available (see *getGFS.m*)

1. **Select a NDVI area**

* Only the listed areas are available

1. **Push the save and exit button**

* The folder structure is generated in the chosen directory
* The file *setup.m* is generated in the folder *app/yourModel/src* containing all relevant information

**How to start the system?**

1. **First installation**

* Setup a directory with RRMDA/src and do the setup as described above.
* Adapt the model name and recipient list in the python script matlab\_oda\_batcher.
* Adjust the parameter files in the folder *RRM\app\yourModel\prm*
* Provide the files *E.mat* and *S0G0.mat* in the folder *RRM\app\yourModel\resources\restart* and the file *samples.mat* in the folder *RRM\app\yourModel\resources\samples* (this should be done automatically in the future!)
* Test run the system with the offline version (no uploading of data on data bases, no sending of mails) with matlab\_oda\_batcher\_offline.py
* Run the file *matlab\_oda\_batcher.py* which was generated by finalizing the GUI

1. **Manual restart**

* Run the file *matlab\_oda\_batcher.py*

1. **Automatic data assimilation on Mac OS X (Yosemite, El Capitan)**The program can automatically run the model and assimilate data. This is managed by the use of a launch daemon.

* Adapt the plist to your needs (i.e. path names)
* Move the plist to /Library/LaunchDaemons
* Open a terminal and type:  
  $ launchctl load /Library/LaunchDaemons/<filename>.plist  
  $ launchctl start <plist\_label>  
  The plist\_label figures in the plist itself or it can be displayed with the following command which lists all daemons containing the name imomowb:  
  $ launchctl list | grep imomowb
* To stop automatic data assimilation type  
  $ launchctl stop <plist\_label>
* To modify the plist it has to be unloaded:  
  $ launchctl unload /Library/LaunchDaemons/<filename>.plist

**Troubleshooting automatic assimilation:** The daemon might stop after system upgrades. After a computer restart do the following checks to make sure the model runs again. Check if the daemon is still active by typing:  
$ launchctl list | grep imomowb  
You can expect an output like:  
- 0 <plist\_label>  
Where the first column contains the id of the process if the daemon is running and the second column is the last return value of the daemon (0 means success). If this output appears there is no need to do anything. If the launchctl command above does not list any imomowb-jobs you have to load the plist of the daemon again with:  
$ launchctl load /Library/LaunchDaemons/<filename>.plist  
You should then find the job with launchclt list | grep imomowb.

**Short description of the most important scripts:**

setupGUI.m

* Sets up the path directory and copies all necessary files
* Catchment organization is specified through the shape file and connection matrix
* Data resources are specified
* Creation of the file master.m containing timer object definition

getRaw.m

* Download of precipitation data from FEWS or TRMM
* Download of temperature data from GDAS
* Download of precipitation and temperature forecast data from GFS
* Download of WMO station data (if available)
* Download of latest eNDVI image (if available)
* Download of latest entries from the iMoMo database

processRaw.m

* Elevation adjustment and bias correction of temperature and precipitation
* Calculation of potential evapotranspiration using Hargreaves equation and crop coefficients obtained by eNDVI
* Downscaling of temperature, precipitation and potential evapotranspiration to sub catchments

runModel.m

* Assimilation of model states and forecasting of hydrological variables

sendtoDB.m

* Send processed data and model results to iMoMo database

matlab\_oda\_batcher.py

* Calls getRaw, processRaw, and runModel / openDA automatically in sequence.