Internal Structure of WaveNet and TideNet

Naveen Ganesh
ERDC CHL
06/18/2019

WaveNet and TideNet Code Composition

- Web2py (Python): Main framework and controller (Web Environment).
- HTML: WaveNet and TideNet interface appearance
- FORTRAN: Modules based on existing code which perform computations
- SQL Lite: Fetching information on a data source.
- Leaflet configured to use ESRI background maps
- Javascript – Google Charts for plotting

Web2Py Composition

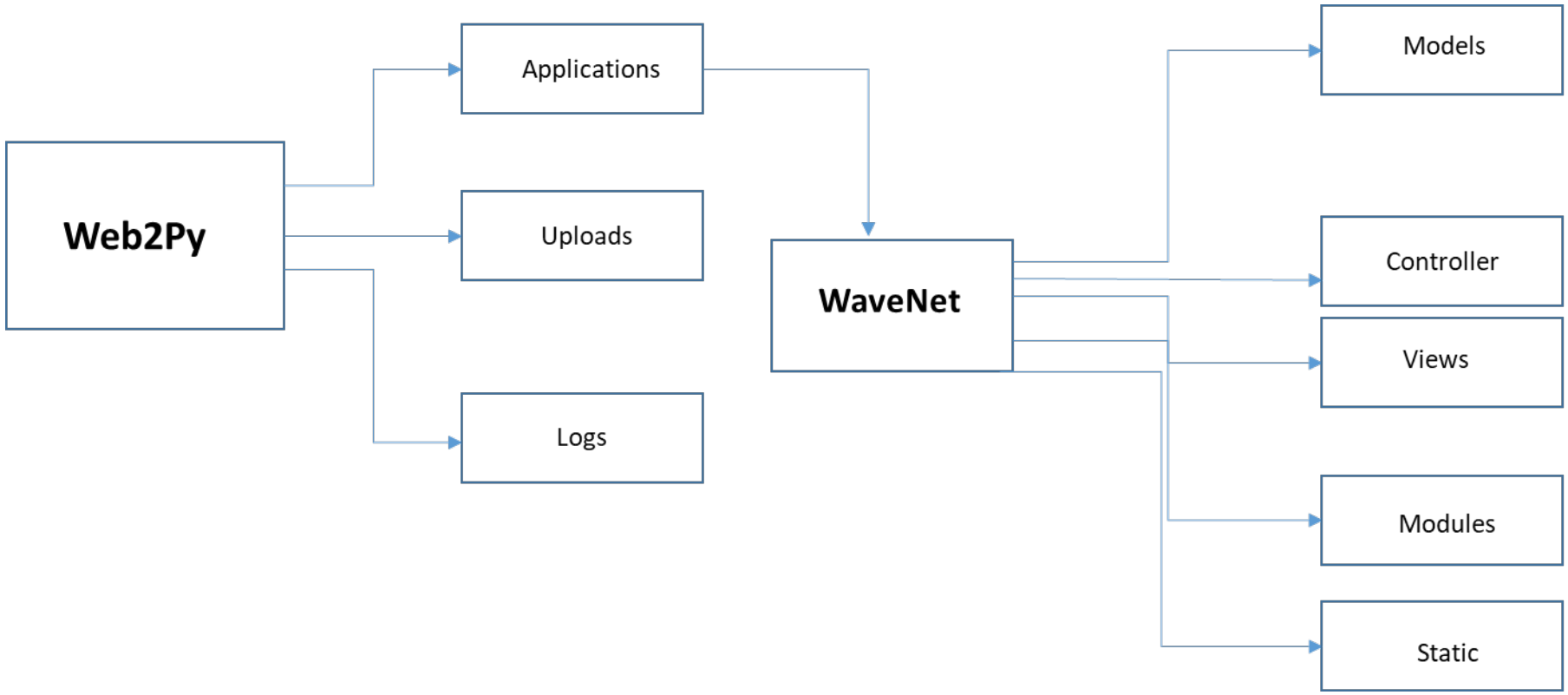
- Web2py consists:

  Applications: WaveNet and TideNet

  Uploads: Contains various files which are downloaded when user does a fetch request, this helps in faster retrieval of data for future requests.

  Logs: Any errors that occurs is reported here, also there is traceback for that error.

WaveNet Overview



Web2py->Applications->WaveNet

- WaveNet in applications consist of:

  Models: WaveNet settings and control of Interface

  Controller: Controls the functionality of WaveNet

  Views: Layout of WaveNet appearance.

  Modules: This consist of code for all the data sources, which help in retrieving data from each data source, also subroutines for plotting is also located here.

  Static: Javascript for displaying maps and other functions of map.

WaveNet->Models

Following are contained in Models:

- Initialization of settings for WaveNet application (0.py)
- Initialization of settings for SQL lite for fetch operation.
- Interface appearance is controlled here (htmlFunctions.py).
- Menu functions in WaveNet is also contained here.

WaveNet-> Controller

- Controller contains following code:

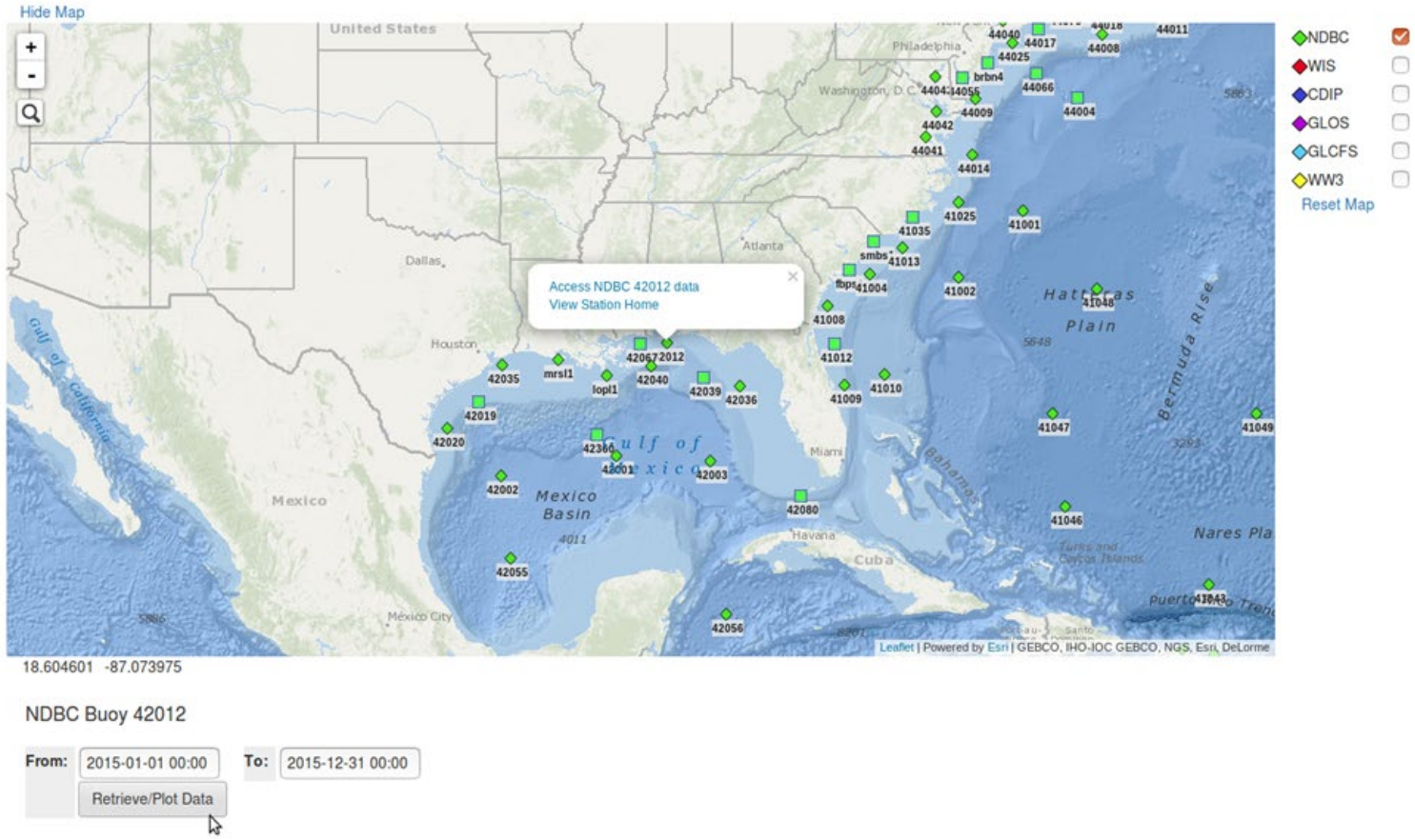
  Code which serve data on to the map (data.py).

  Code which calls routines to create plots (roseplt.py).

  Code which retrieves data on user request, deals with tabular data.

  Lastly code that helps in generating statistics.

WaveNet-> Controller

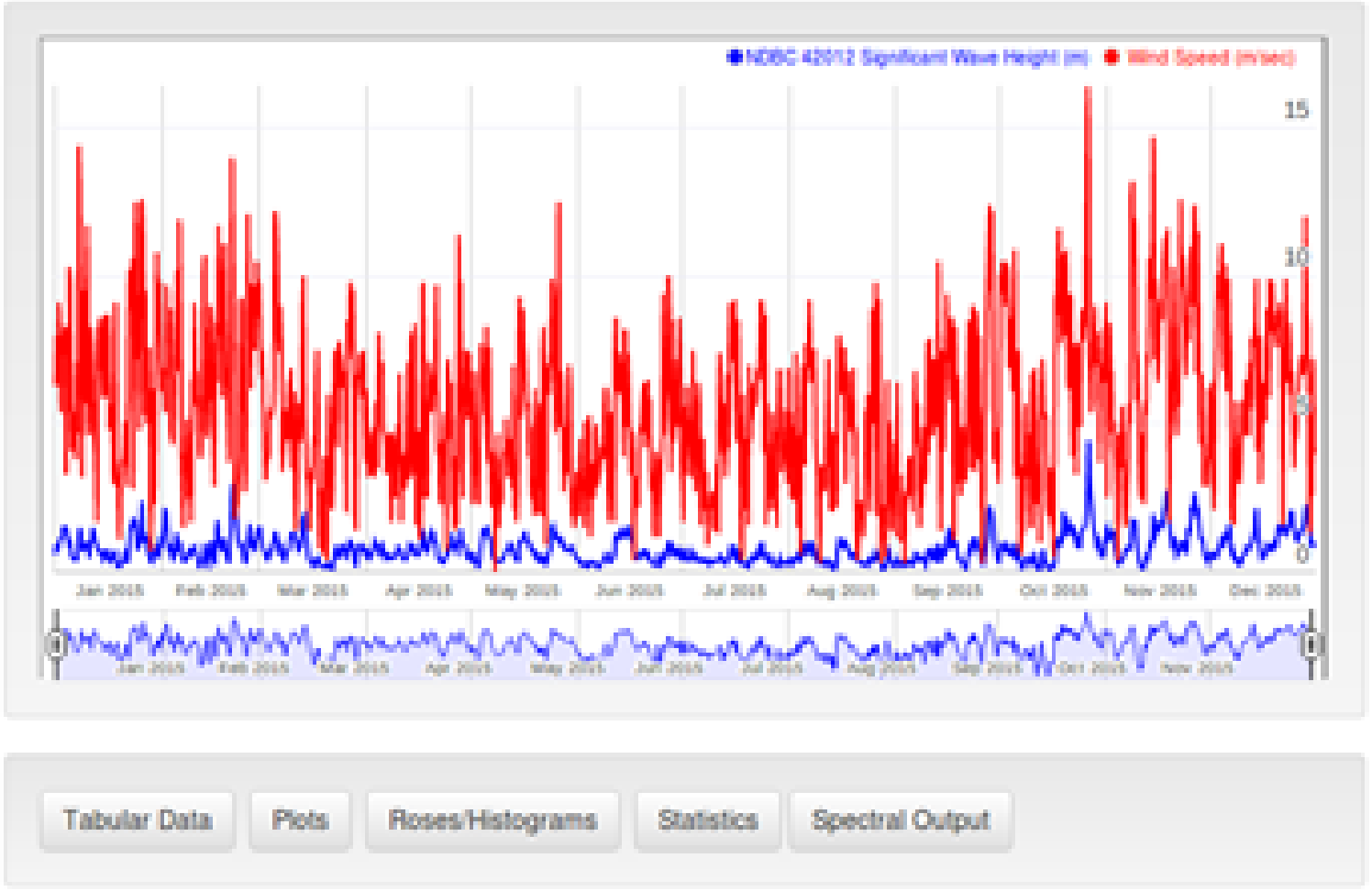


Code in Controller populating the map with data and fetch operation

WaveNet->Modules

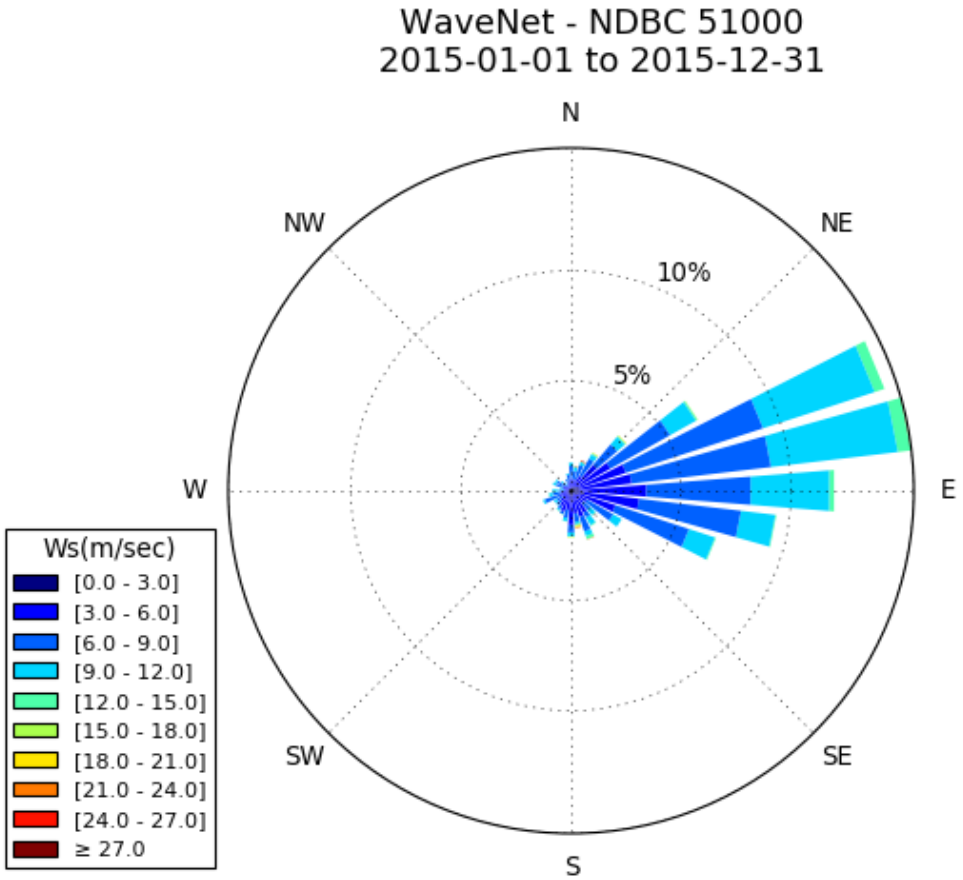
- Modules in WaveNet contains code for functionality of each data source, whenever the user requests data and plot.
- The data sources present here are CDIP, COOPS, GLCFS, GLOS, NDBC, WIS , WvW.
- Along with code for data retrieval for each of the data sources, there is code which helps in plotting timeseries and wind rose plots.

WaveNet->Modules



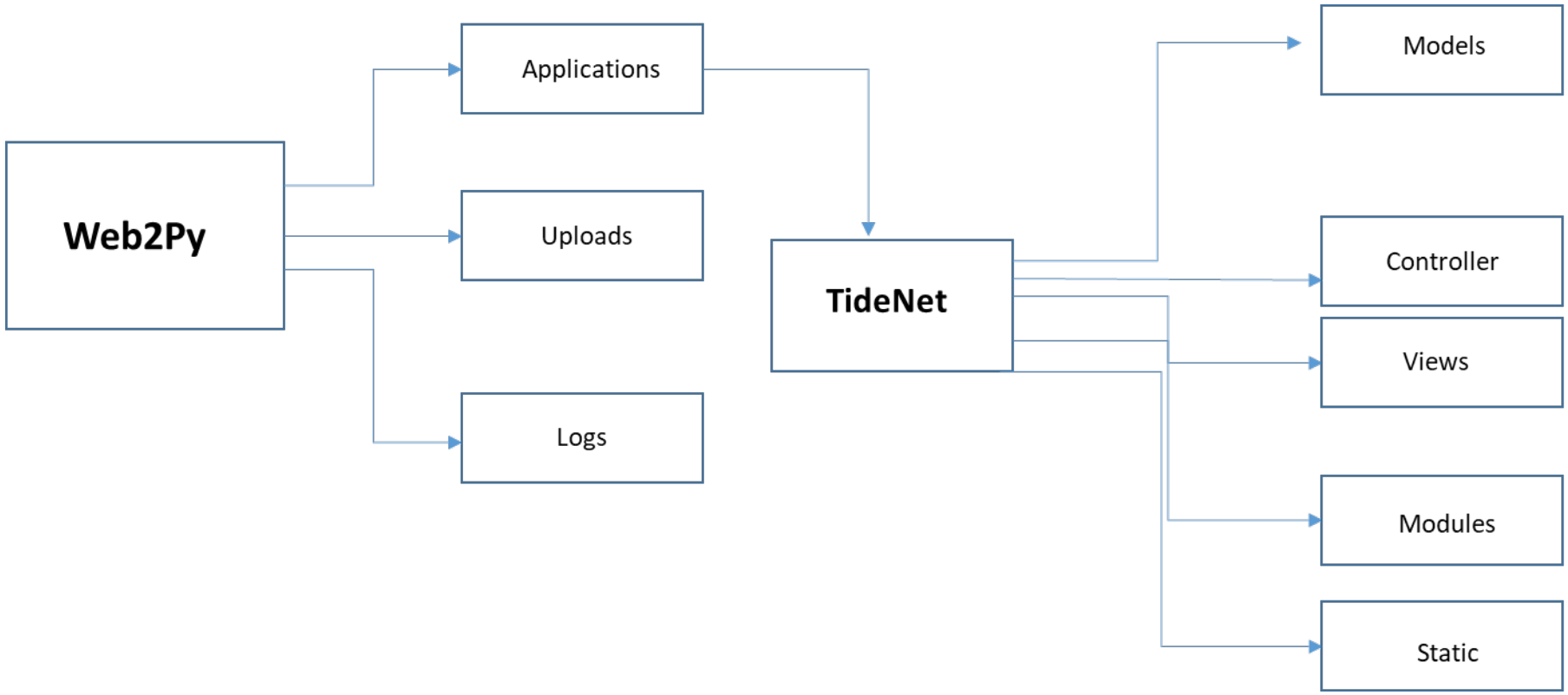
Code in Modules which creates time series plot from the data source

WaveNet->Modules



Code in Modules which creates the wind rose plot from the data source

TideNet Overview



Web2py->Applications->TideNet

- TideNet in applications consist of:

  Models: TideNet settings and control of Interface

  Controller: Controls the functionality of TideNet

  Views: Layout of TideNet appearance.

  Modules: This consist of code for all the data sources, which help in retrieving data from each data source, also subroutines for plotting is also located here.

  Static: Javascript for displaying maps and other functions of map.

TideNet->Models

Models similar in WaveNet contains the following:

- Initialization of settings for TideNet application (0.py)
- Initialization of settings for SQL lite for fetch operation.
- Interface appearance is controlled here (htmlFunctions.py).
- Menu functions in TideNet is also contained here.

TideNet-> Controller

- Controller similar in WaveNet contains following code:

  Code which serve data on to the map (data.py).

  Code which calls routines to create plots (roseplt.py).

  Code which retrieves data on user request, deals with tabular data.

  Lastly code that helps in generating statistics.

TideNet->Modules

- Modules in TideNet again like WaveNet contains code for functionality of each data source, whenever the user requests data and plot.
- The data sources present here are ADCIRC, COOPS, LEPRO, OSU.
- Along with code for data retrieval for each of the data sources, there is code which helps in plotting time series and wind rose plots.

Demo