Deltaces Enabling Delta Life

FFFS – Future Flood Forecasting System

Ivo Miltenburg (Marc van Dijk, Martijn Kwant)

\Rightarrow

Outline – Future Flood Forecasting System (FFFS)



Andrew Gubbin Director

Nadine Slootjes Department Head **Environment Agency Project Director**

Gerben Boot **Delft-FEWS Product** Manager

Marc van Dijk **Project Manager** **Environment Agency Project Manager**

Onno van den Akker Lead Developer

Jan Verkade SME

Environment Agency Solution Architect

Rudie Ekkelenkamp Developer

Paul Wass SME

Environment Agency SME

Erik Pelgrim Developer

Alex Minett SME

Environment Agency SME

Ruth Preator Developer

Lora Buckman SME

Anne Hommelberg Junior Developer

Martijn Kant SME

Deltares Staff

Marc van Dijk: project lead

Project team

Onno van den Akker: lead developer

•Jan Verkade: fluvial/coastal configuration

•Lora Buckman: BWQFS configuration

•Rudie Ekkelenkamp: developer (Azure, services)

•Martijn Kwant: Delft-FEWS configuration scripting

Dave Casson: Delft-FEWS configuration scripting

JBA Staff

Alex Minett, Paul Wass, Rob Berry, ...



Deltares

NFFS – FFFS Project History

In 2003 National Flood Forecasting System for England and Wales

- Migrate existing regional systems to Delft-FEWS client-server
- Large differences in models and experience (>20 models)
- Harmonisation: national consistency and National Datasets

In 2007: National implementation for FFC

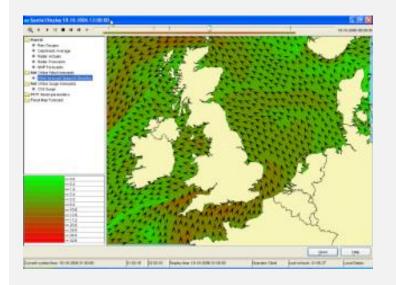
- Close cooperation between EA and Met Office
- Use of new ensemble products for probability forecasting

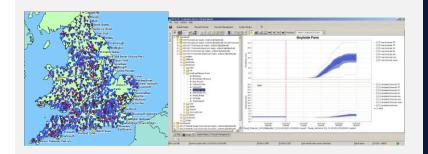
In 2010: National **Bathing Water Quality** System

- National Coastal lookup tables for water quality
- Use observed and forecasted rainfall as input









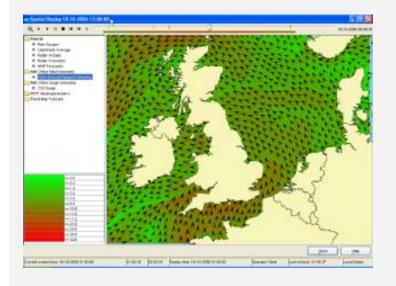
NFFS – FFFS Project History

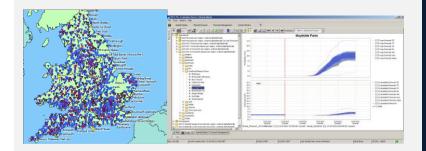
2018-2019: Future Flood Forecasting System

- One national system for floods & bathing water quality
- Simplified maintenance: **scripted configuration** from reference database
- Deployment in the cloud (Azure)



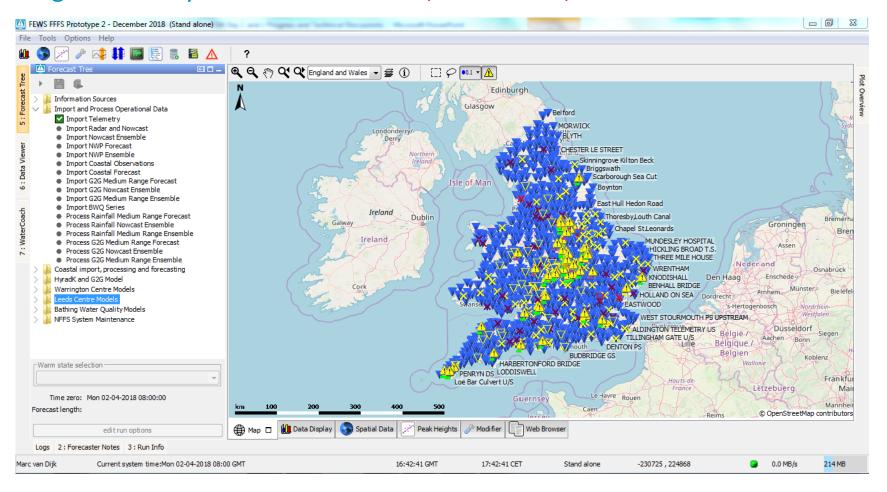






Future Flood Forecasting System (FFFS)

- One system for 7 Regional Centres, FFC and BWQ
- Three integrated sub-systems: R&T database, Delft-FEWS, Web Portal





Current Project and Planning

Deltares made a plan with JBA for software developments and national implementation

- Show new Delft-FEWS functionality in 3 prototypes
- Script the Delft-FEWS configuration and use template functionality where possible
- Provide client-server application in Deltares estate
- Develop Reference and Threshold database with all FFFS meta and model data
- Develop Python scripts to script the Delft-FEWS configuration
- Develop new Delft-FEWS functionality
- Configure a National system with NFFS, FFC and BWQFS
- Install system on Azure cloud







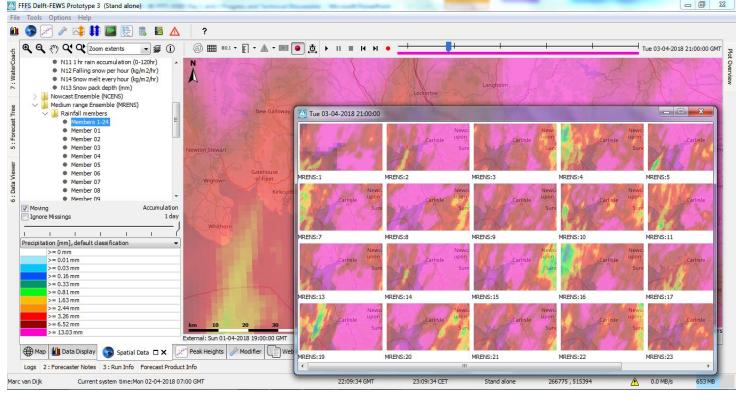
FEWS Developments for FFFS

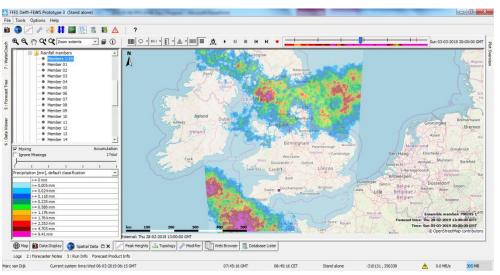
There are many Delft-FEWS developments planned/realised:

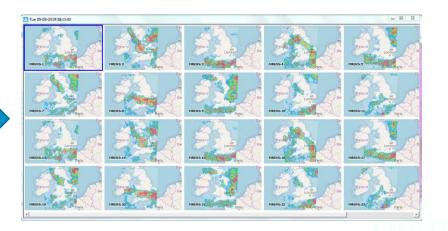
- (Better integration with Microsoft Azure cloud)
- (New Delft-FEWS dashboard display)
- Postage Stamps in Spatial Display
- Spatial Modifier
- Sys Admin developments
- New Threshold Crossing Display / Multiple
 Threshold groups
- Forecast Product Information
- Creating a Delft-FEWS config from a database (scripted configuration)



FEWS Developments: Spatial Display Postage Stamps



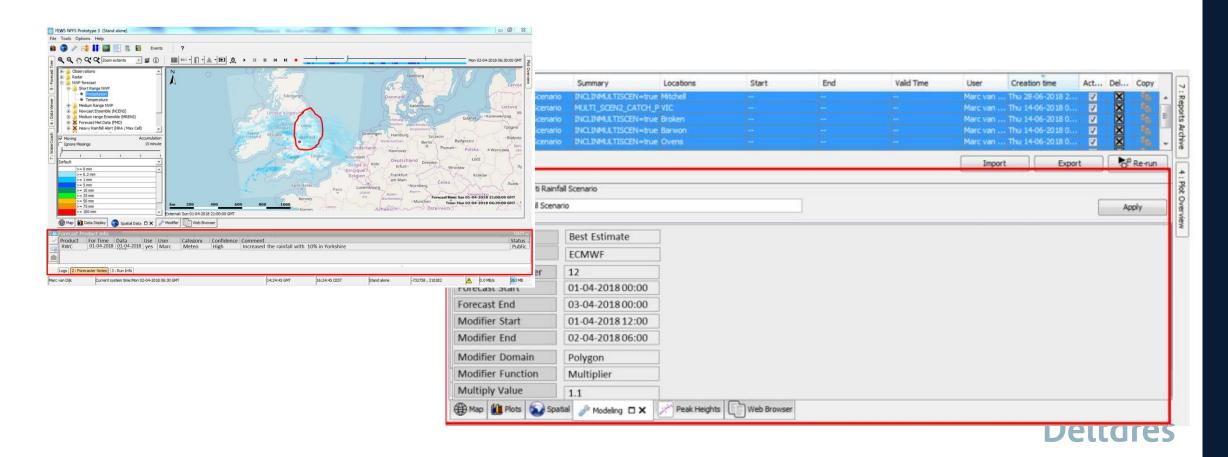




Delft-FEWS Software Developments:

Spatial Modifier

Modifier linked with Spatial Display for grids



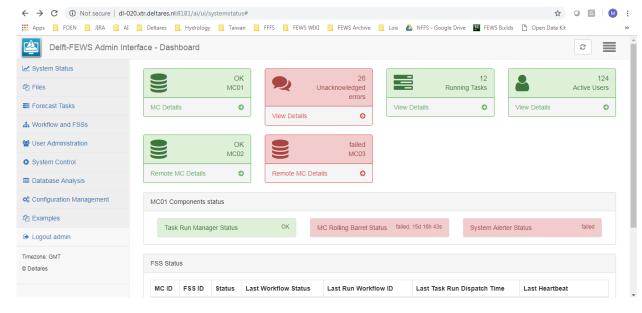
Delft-FEWS Software Developments

Admin Interface Web Service

Admin Interface

For making the system admin information available to external applications a new Delft-FEWS Admin Interface Rest Web Service is being developed. On top of this Admin Interface Rest Web Service a new Admin Interface is being developed

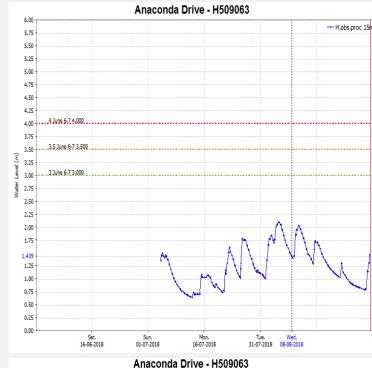
- 1. New design
- 2. Responsive: works on tablets/phones as well
- 3. All functionality available as REST API
- 4. Upload base build / patch
- 5. Security groups: Al_ADMIN and Al_VIEWER
- 6. Map workflows on matrix page

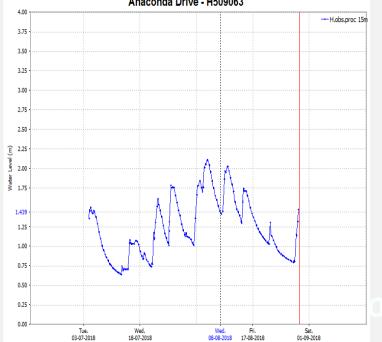


Delft-FEWS Software Developments

Seasonal Thresholds

- Thresholds have a validity period (e.g. a season)
- Viewperiod graph determines visibilityDefined in thresholdValueSets (e.g. <season startMonthDay="0103" endMonthDay="3006"/>)
- Threshold icon is also affected.
- Functionality based on system time

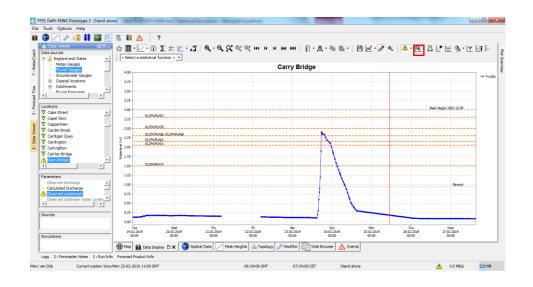


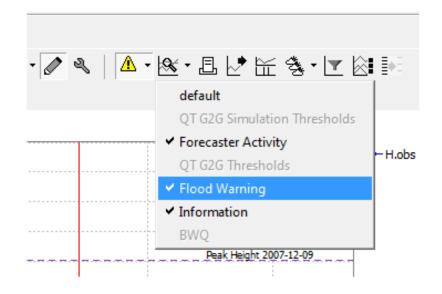


Delft-FEWS Software Developments

• Define and visualize multiple thresholds

▲ thresholdGroup (3)					
an conoladi da	9 (0)	= id	= name	() defaultThreshold	() levelThreshold
	1	Flood_Warning	Flood Warning	■ defaultThreshold shortName=RES	■ levelThreshold (1)
	2	Forecaster_Activity	Forecaster Activity	■ defaultThreshold shortName=ACT	▼ levelThreshold (1)
	3	Information	Information	defaultThreshold shortName=Info	▼ levelThreshold (1)
Au Commont		C2C OT Thropholds			





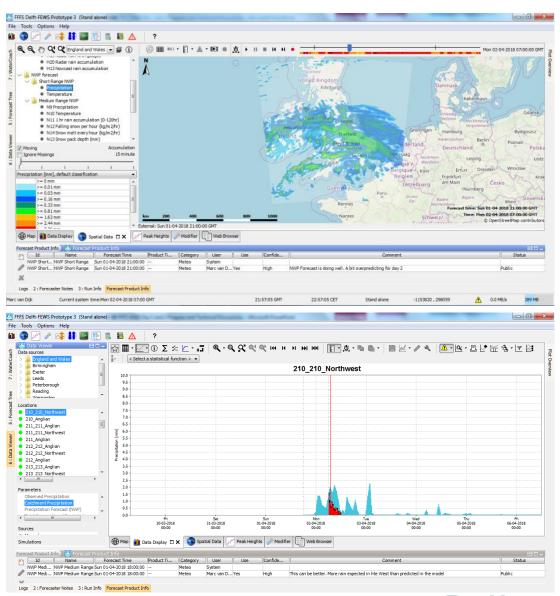
Multiple groups can be selected



Delft-FEWS Software Developments

Forecast Information Display Developments

- Forecast Information Display is being developed.
- Forecast product comments can be entered, edited and deleted
- Time series display can show comments (series and comments are connected)



Creating a config from a database

One of the requests for the new Future Flood Forecasting System in the UK was to to minimalize XML/FEWS configuration efforts.

As a result it was decided to store all relevant meta-data of the FEWS configuration in a SQL server database and automatically generate a FEWS configuration from this database.

This led to several design decisions:

- All relevant information (meta-data) for FEWS is stored in a (Azure SQL) database, e.g.
 locations, thresholds, workflows and model information → a single centralized place where all
 information is stored!
- A GUI will let the end-user easily edit and update information in the database
- The generated config will exists of a static part (templates) and an updated part





National consistent Delft-FEWS configuration

- We want to script the FFFS Delft-FEWS configuration as much as possible
 - Challenge for the EA to make sure their procedures are national consistent
 - Consistency in processing of rainfall data, data hierarchy, type of models, ...
- Because the project lead is with the Regional fluvial forecasting teams:
 - Fluvial regional models are National consistent
 - Coastal procedures and models are (not yet) consistent
 - FFC models and procedures are not consistent
 - BWQFS locations and procedures are not consistent
- The different regional systems have many small modules that are still not national consistent (Thames barrier, Kflows, Regression, ..).





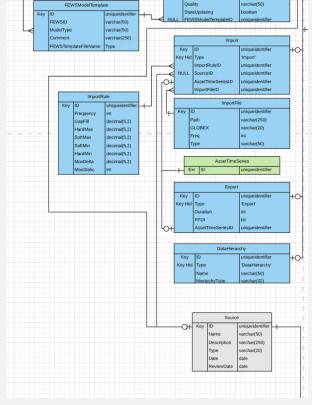
Creating a config from a database

Using a database has several advantages:

- One centralized place to store all FEWS metadata, no issues with different versions
- 2. Allows a user to autmatically generate a FEWS configuration (e.g. export location tables from the Database)
- 3. Due to the linkages, related configuration is automatically updated
- 4. Automatically perform checks to ensure data validity. (e.g. an ID cannot contain spaces)



Database



Scripting a National FFFS Delft-FEWS System

- EA (and JBA) is creating a data model that contains most of the reference data for FFFS
- JBA is responsible for populating the data model to the Azure SQL server database and develop a User Friendly GUI. Forecasters will use the GUI to change the FEWS configuration.
 - Gauge locations with additional attributes (Id, Name, X, Y, Z, Region, Catchment, Type, ..)
 - Catchments with additional attributes (Polygon, Area, ID, Code, ..)
 - Information to exchange data with other systems (Telemetry, WISKI, MO, Warning, ..)
 - Processing information (Rain-gauge weightings, Ratings)
 - Information for modelling (model type, linkage of models to rivers and catchments ..)
 - Information for **forecasting** (Thresholds, Warning, ..)



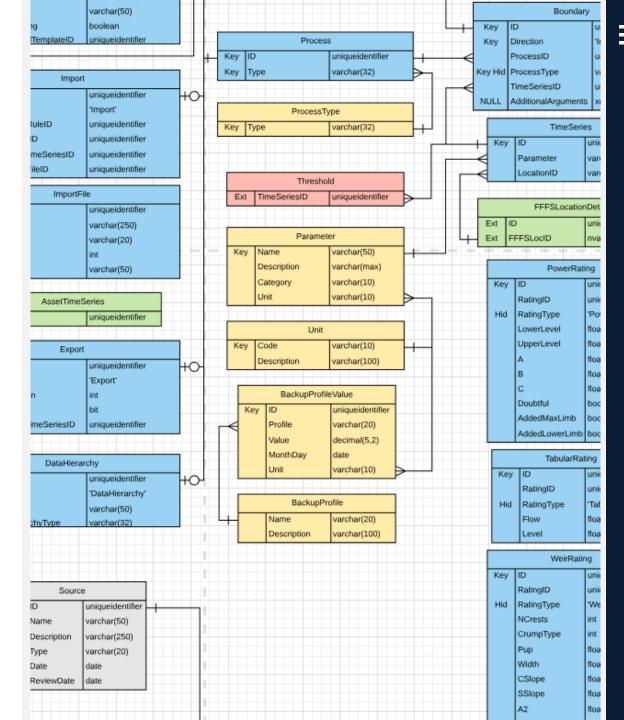


19

Scripting a National FFFS Delft-FEWS System

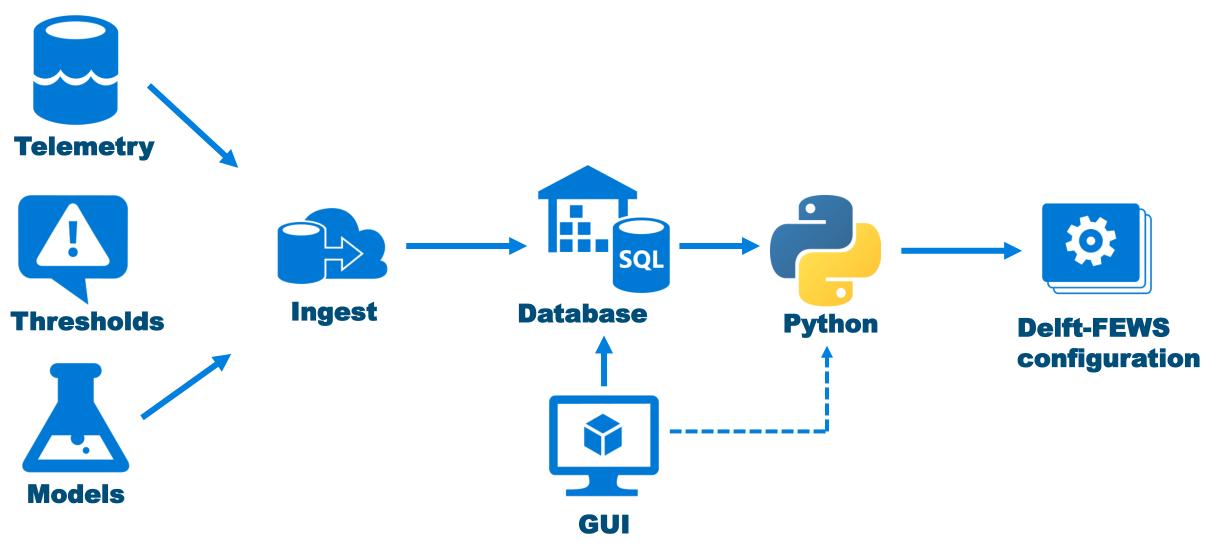
Can this EA data model be used directly by FEWS for FFFS: No!

- Data from the different database tables will need to be converted to a 'FEWS friendly' structure and format.
- We need to write scripts to convert the EA data model to this 'FEWS friendly' format



Deltare

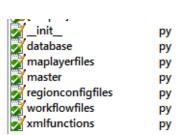
Scripting a FEWS-FFFS system from a database

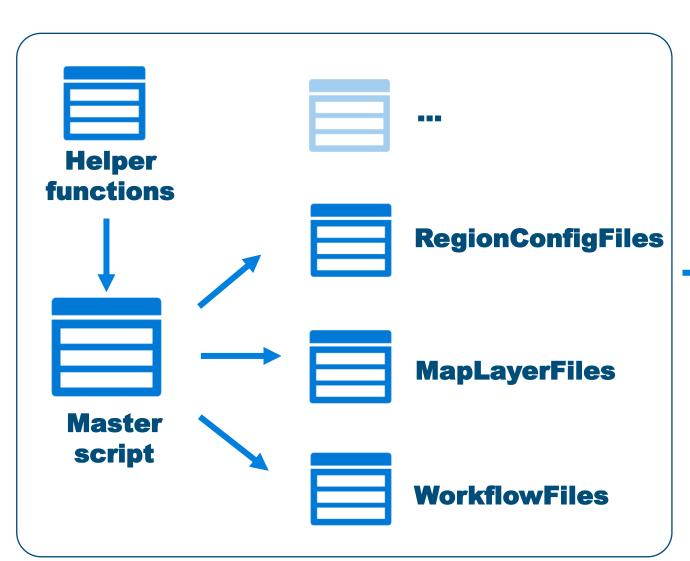


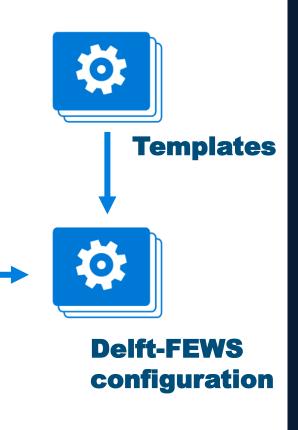
Deltare

Creating a FEWS-FFFS data model from EA data model









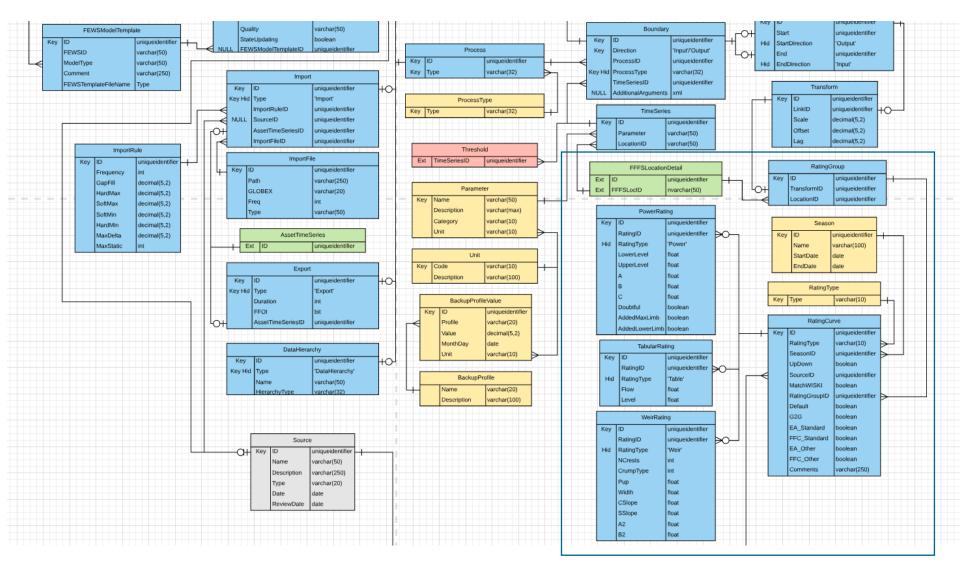
Deltares

SQL queries

- Getting data out of the database: SQL (Standard Query Language)
 - SQL is basically a way of telling where (in which tables) to find what information
 - And how different tables are connected to each other
- SQL is the link between the Python scripts and the Database
- Important part of generating the FEWS config

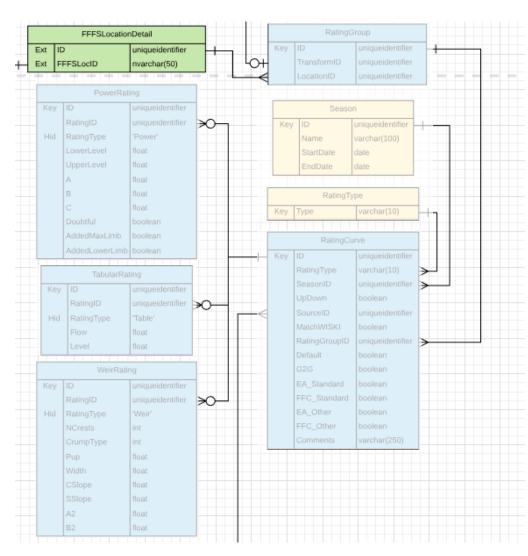
```
sql = 'SELECT FFFSLocationDetail.FFFSLocID, Location.Name, TabularRating.Flow,\
TabularRating.Level FROM [fews].[TabularRating] Left JOIN [fews].[RatingCurve]\
ON [fews].[TabularRating].RatingID = [fews].[RatingCurve].ID LEFT JOIN \
[fews].[RatingGroup] ON [fews].[RatingGroup].ID = [fews].[RatingCurve].Rating\
GroupID INNER JOIN [dbo].[FFFSLocationDetail] ON [fews].[RatingGroup].\
LocationID = [dbo].[FFFSLocationDetail].ID INNER JOIN [internal].[Location] \
ON [internal].[Location].ID = [dbo].[FFFSLocationDetail].LocationID'
```

Rating Curves example

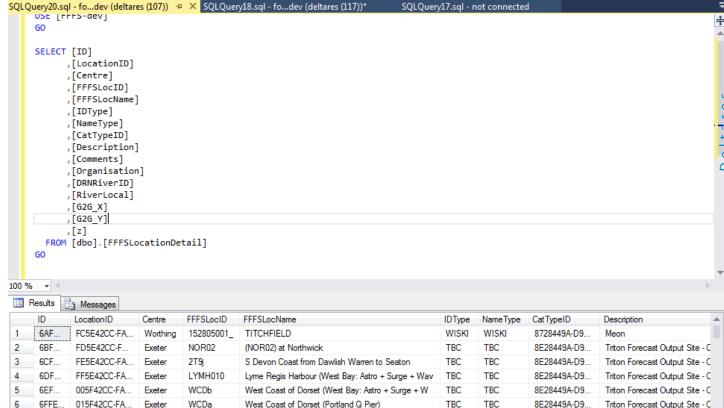




Rating Curves and SQL



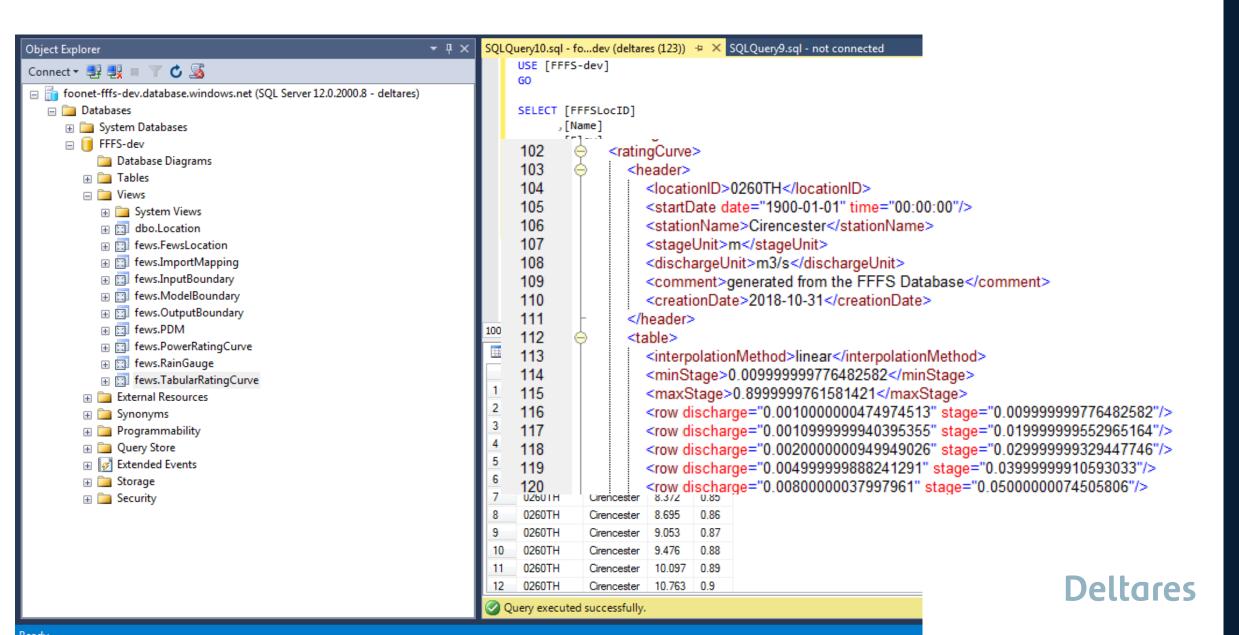
INNER JOIN [dbo].[FFFSLocationDetail] ON [fews].[RatingGroup].\
LocationID = [dbo].[FFFSLocationDetail].ID

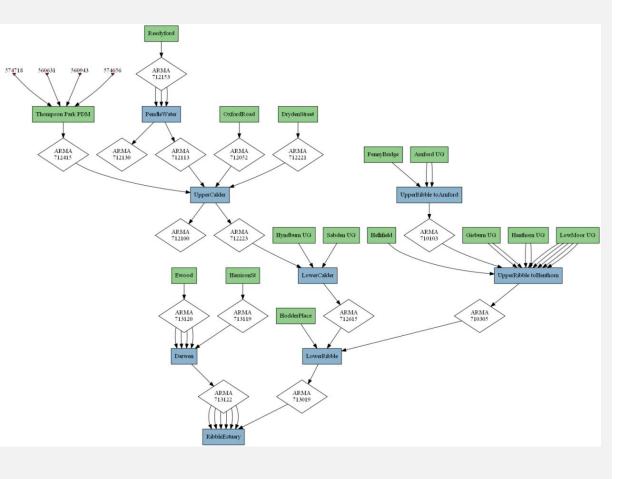


Example script to write XML

```
12 class RatingCurves:
13
14
      def init (self):
                                                                                                                                      RegionConfigFiles
15
16
          self.outdir = os.path.join(os.getcwd(), 'output')
17
          self.configdir = xf.create subdir(self.outdir, 'Config')
          self.dir = xf.create_subdir(self.configdir, 'RegionConfigFiles')
18
19
          self.relations = db.get relations()
          self.startdate = {'date':'1900-01-01', 'time':'00:00:00'}
20
          self.stageunit = 'm'
22
          self.dischargeunit = 'm3/s'
23
          self.creationdate = '2018-10-31'
24
25
26
      def add header(self, child1 ,loc id, loc name):
          '''add header'''
27
28
29
          child1A = ET.SubElement(child1, 'header')
30
          xf.add child with text(child1A, 'locationID', loc id)
          ET.SubElement(child1A, 'startDate', self.startdate)
31
32
          xf.add child with text(child1A, 'stationName', loc name)
          xf.add child with text(child1A, 'stageUnit', self.stageunit)
33
81
82
      def tabular rating curves(self, root):
           ''' This method calls a dataframe containing the tabular rating curves and adds them to an XML file'''
83
84
          df = db.read_data('SELECT [FFFSLocID], [Name], [Flow], [Level] FROM [fews].[TabularRatingCurve]')
85
          df = df.sort_values(['FFFSLocID', 'Flow'], ascending=[1 ,1])
          self.add ratingcurve(root, df, 'table')
87
88
89
90
      def power rating curves(self, root):
              This method calls a dataframe containing the powerequation rating curves and adds them to an XML file'''
91
92
          df = db.read data('SELECT [FFFSLocID],[Name],[SeasonID],[RatingGroupID],[SourceID],[LowerLevel],[UpperLevel],[A],[B],[C] FROM [dbo].[PowerRatingCurve]')
93
          df = df.sort values(['FFFSLocID', 'LowerLevel'], ascending=[1 ,1])
          df = df.drop duplicates()
          self.add ratingcurve(root, df, 'power')
```

Example of a view in the database





Automatically generating Workflows

The database also contains linking of different models, e.g. precipitation-discharge models (green), ARMA modules (white) and hydrodynamic models (blue).

These schematics are used to generate workflows and the Forecast Tree

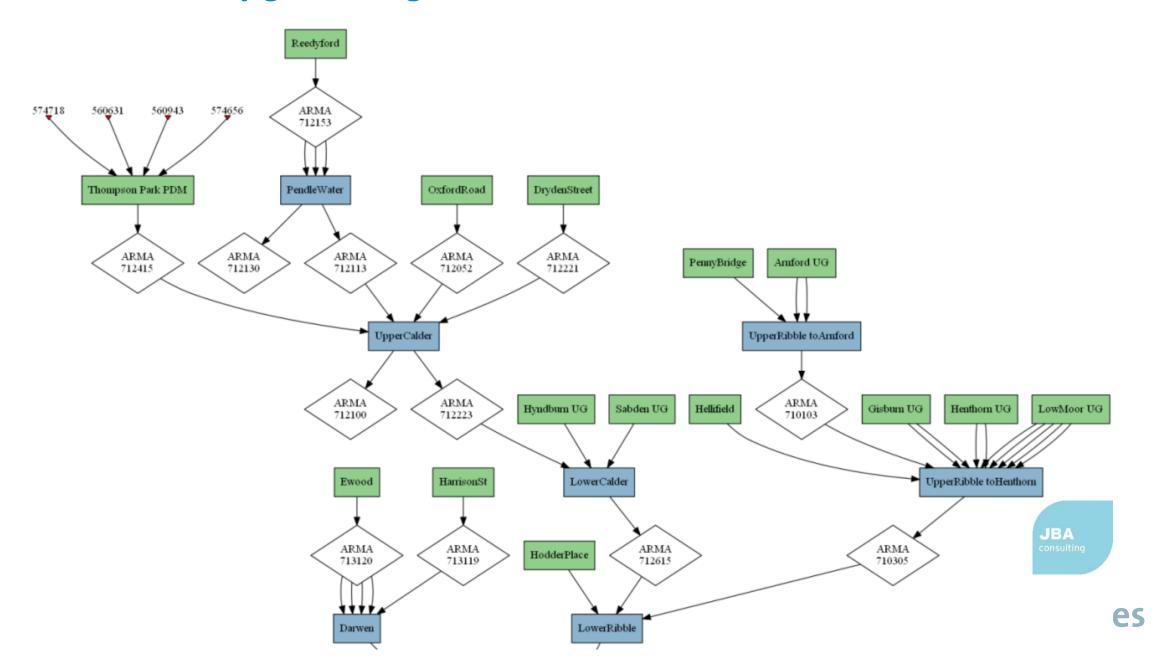


Automatically generating Workflows

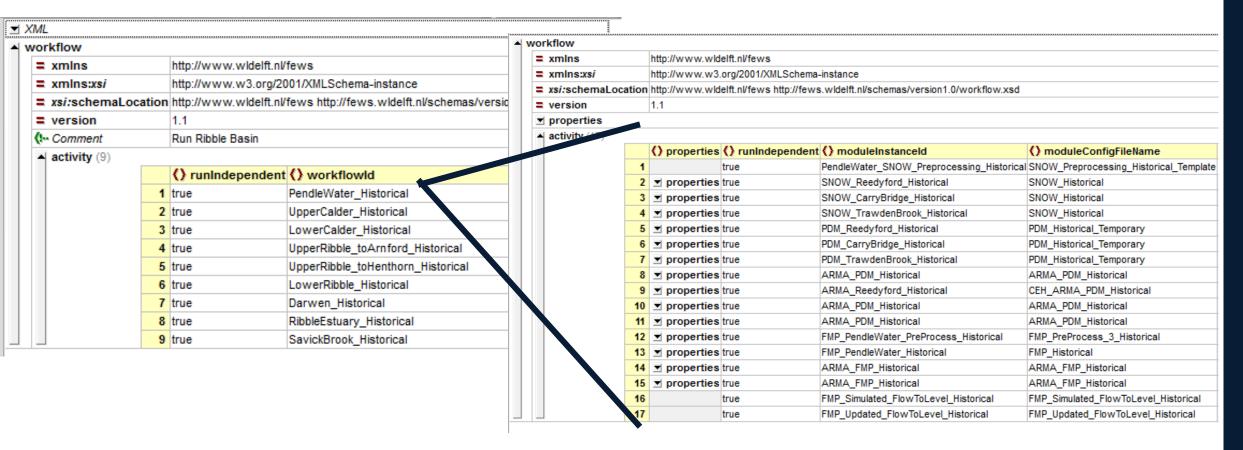


- Automatically generating workflows
 - > Database contains intelligence of the linking between models
 - > Used to create a schematic per basin of all the models and how they are connected
 - > This information is then used to automatically generate the workflows and forecast tree
 - > Workflows are templated: The script will check:
 - 1. What the input properties of the model are
 - 2. Which type of model needs to be run
 - 3. What the output properties are
 - > This is used as input for the template

Automatically generating Workflows



Scripted Workflows for FFFS





FFFS Project plan and proposal

All developments, configuration activities, milestones, training, documentation included in MS project

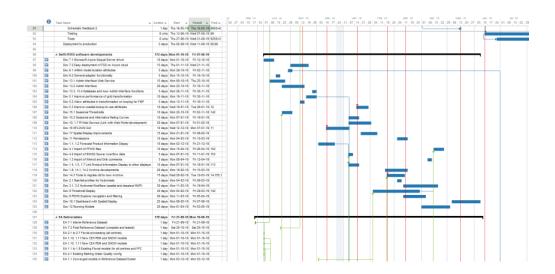
Many dependencies between EA deliverables, FEWS developments, JBA developments, FEWS

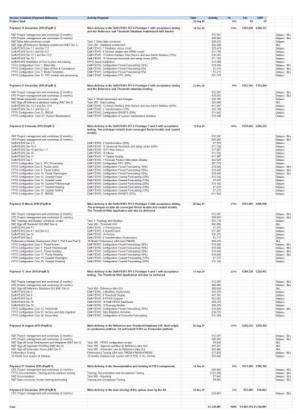
configuration activities.

All developments and configuration must be completed by August 2019

September-December 2019: parallel running of NFFS with FFFS

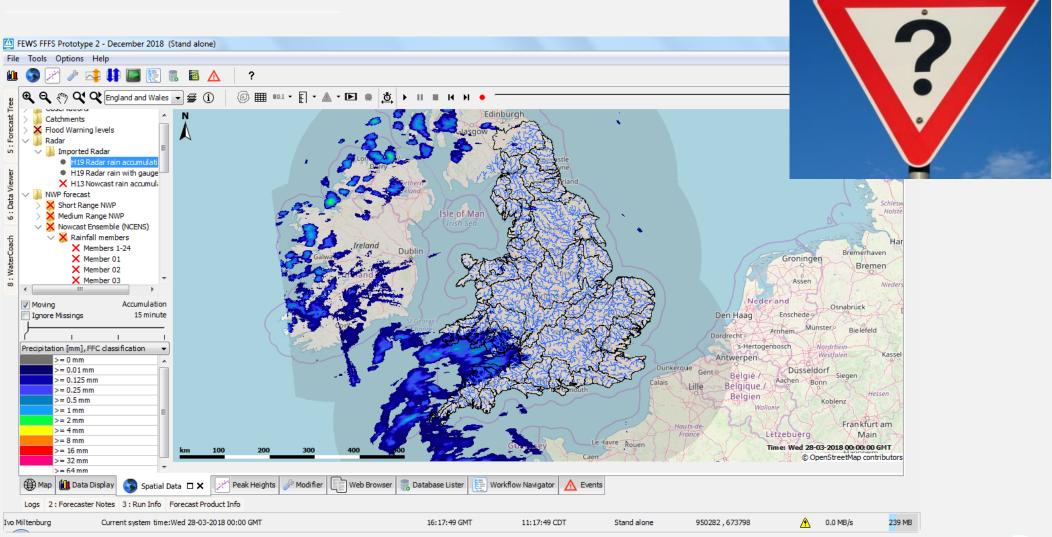
1 January 2020 switch to new FFFS system







Questions





Deltares

Thanks for your attention

#