

# Overview of the Delft-FEWS platform

By Susana Ochoa-Rodriguez, Imperial College London  
([s.ochoa-rodriquez@imperial.ac.uk](mailto:s.ochoa-rodriquez@imperial.ac.uk))

Compilation of material prepared for RainGain's WP3 technical meetings held at Imperial College London and KU Leuven between June 2012 – August 2013



# Some existing Flood Forecasting Platforms

## (Flood Early Warning Systems - FEWS)

- In-house linkage of input data and models - Tailored for particular needs, research projects
- UrbanFlood Common Information Space (CIS): open platform for Early Warning Systems (EWS)
- Innovyze Floodworks
- Delft-FEWS
- ...

All have similar structure, the final purpose is the same



- **Open Source, Open shell** system for managing forecasting processes and/or handling time series data
- Collection of modules designed for building a hydrological forecasting system customised to the specific requirements of an individual organisation
- User-friendly GIS interface, wide range of tools for visualisation, analysis and handling of data
- Provides open interface to any external forecasting model and/or data handling routine



# Why Delft-FEWS?

- Open Source
- Flexible, extensible modular structure
- Used for simple as well as complex applications
- Supports wide range of standard formats
- Built-in tools
- Has been widely tested – operationally used in more than 20 countries: US, UK, Taiwan, NL, Singapore, etc.
- Complete documentation
- User forum
- We have some expertise and help from Deltares

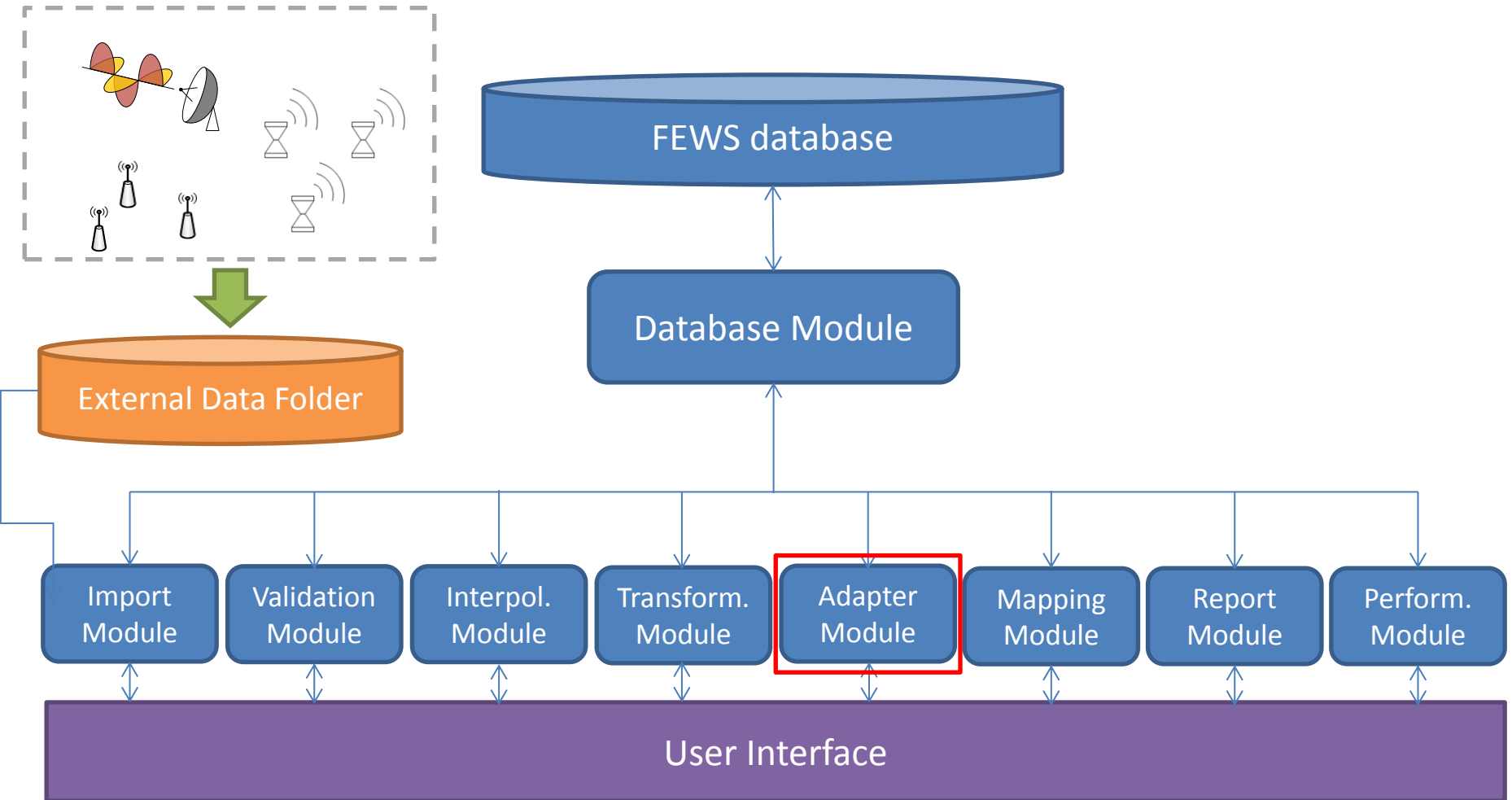


# More about Delft-FEWS

- Developed using Java
- Fully configurable through open XML formatted configuration files
- Data are imported using standard interchange formats (XML, CSV, GRIBB, ASCII, etc.)
- Also supports import of ensemble weather predictions

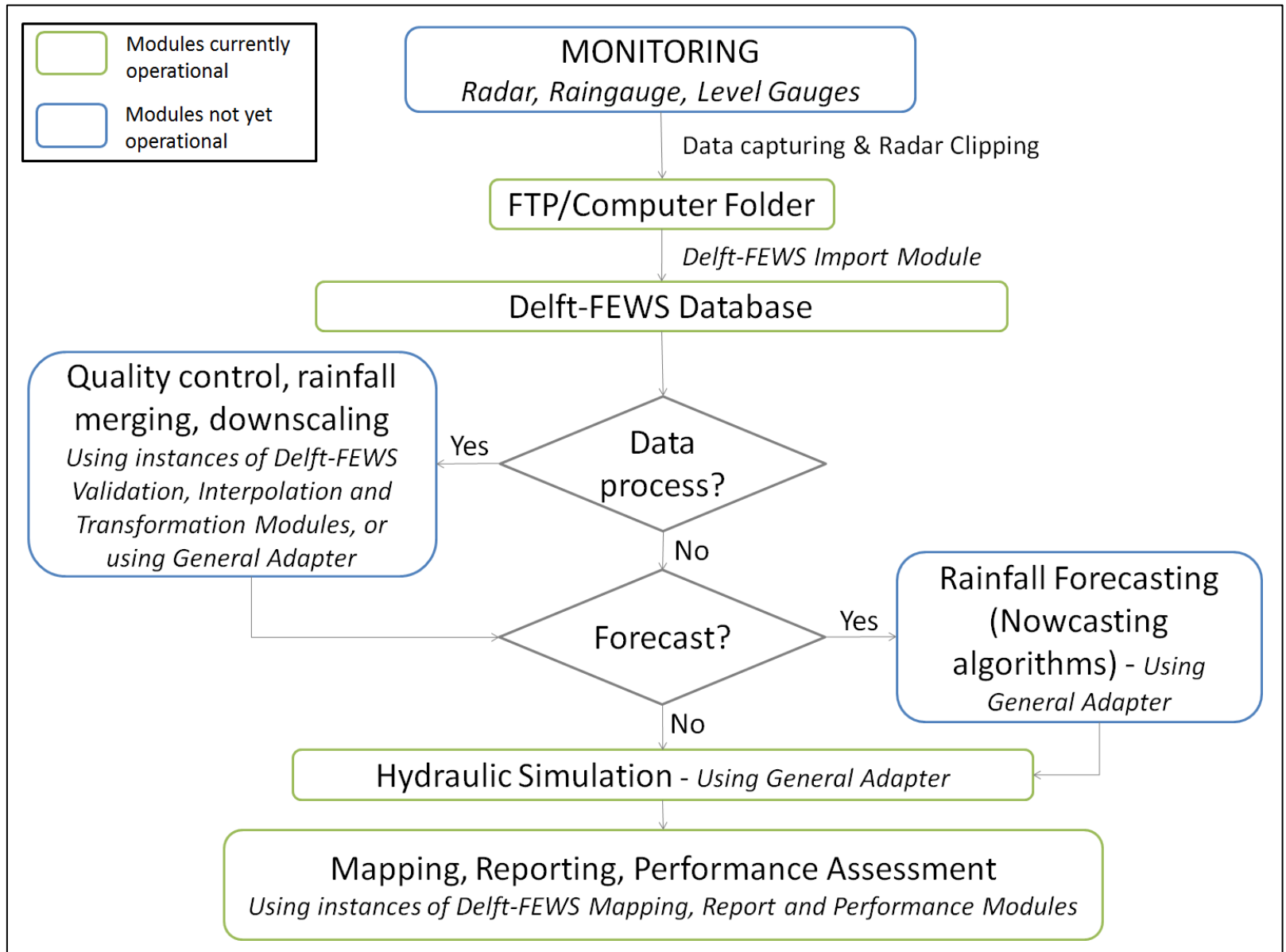


# Delft-FEWS Modular System



As can be seen, it has a central database to which a number of modules with different functionalities can be plugged. One of these modules is the “Adapter”, which enables linking external models to the system (e.g. algorithms for rainfall nowcasting and downscaling, hydraulic models, amongst others).

# Flow of Information



# Proposed Setup

- Delft-FEWS can be run in two ways:
  - As a self-contained (standalone) manually driven forecasting system operation on a normal desktop computer
  - As a fully automated distributed client-server application
- Proposed setup:
  - Standalone version installed on a desktop computer located in pilot leaders' workplace
  - In this way we avoid confidentiality and communication issues





- The Delft-FEWS system was developed using Java (thus the need of installing the JRE), but it is **fully configurable through open XML** (Extensible Markup Language) formatted configuration files.
- In XML files, **tags** are used to mark the start and end of **elements**, which are the logical units of information in an XML document.

*< geoDatum >* *WGS 1984* *</geoDatum >*  
*start tag* *actual content* *end tag*



**Module**

**vs.**

**Instance of a  
Module**



General type



Specific object of  
the type

Class



Object

Humans



- Delft-FEWS has multiple modules to perform a number of activities



- When you want to carry out one of these activities, you must **create a new instance of a module**. This instance must be **customised** to your specific needs (e.g. type of data that you will handle, locations, folders from where to get and where to output data, etc.).



# General steps for creating and using a new module instance (applies to all modules, except General Adapter)

## 1. Design your new instance:

- What will it do?
- Inputs?
- Outputs?

## 4. Create and set folders to be used by new instance:

- Create input and output folders anywhere you want
- Specify location of folders of this instance in:  
*FEWS\FEWS\_SA\sa\_global.properties*

## 5. Check (or create) locations and parameters to be handled by new instance in:

- FEWS\FEWS\_SA\Config\RegionConfigFiles\Parameters.xml*
- FEWS\FEWS\_SA\Config\RegionConfigFiles\Locations.xml* OR *LocationSets.xml* AND for grid data: *Grids.xml*

## 2. Create new module instance configuration file (\*.xml):

- Use template according to the module that the instance belongs to
- This xml file can take any name
- Place new xml file in:  
*FEWS\FEWS\_SA\Config\ModuleConfigFiles*
- Includes all details of the instance

## 6. Create new Map ID file (\*.xml) for new instance:

- This file maps external data and locations to internal parameters and locations
- Use as template existing Map ID files
- Place new xml file in:  
*FEWS\FEWS\_SA\Config\IdMapFiles*

## 7. Create a workflow (\*.xml) which allows executing the new instance:

- Place new workflow xml in:  
*FEWS\FEWS\_SA\Config\WorkflowFiles*

## 8. Register new workflow:

- In: *FEWS\FEWS\_SA\Config\RegionConfigFiles\WorkflowDescriptors 1.00 default.xml*

## 3. Register the new instance:

- In: *FEWS\FEWS\_SA\Config\RegionConfigFiles\ModuleInstanceDescriptors 1.00 default.xml*
- Need to include:
  - Name of instance
  - ID of module it belongs to
  - Subjective description

## 9. Design display of data generated by new module instance:

- If time series:  
*FEWS\FEWS\_SA\Config\SystemConfigFiles\DisplayGroups 1.00 default.xml*
- If spatial data:  
*FEWS\FEWS\_SA\Config\DisplayConfigFiles\SpatialDisplay 1.00 Default.xml*

## KEY

- **Design activity**
- **New file needed**
- **Modification to existing file**

# General steps for creating and using a new instance of the General Adapter (GA) module

## 1. Design your new instance:

- What will it do?
- Inputs?
- Outputs?

## \*2. Create and set a new folder for executable file

- Create new folder in:  
*FEWS\FEWS\_SA*
- Place the executable file in the new folder
- Specify the location of the folder in:  
*FEWS\FEWS\_SA\sa\_global.properties*

## 3. Create new module instance configuration file (\*.xml)

- Using template of GA module
- This xml file can take any name
- Place new xml file in:  
*FEWS\FEWS\_SA\Config\ModuleConfigFiles*
- Includes all details of the instance, **including arguments of algorithm (e.g. location of input/output folder)!**

## 4. Register the new instance

- In:  
*FEWS\FEWS\_SA\Config\RegionConfigFiles\ModuleInstanceDescriptors*  
*1.00 default.xml*
- Need to include:
  - Name of instance
  - ID of module it belongs to (**GA**)
  - Subjective description

## 5. Create and set input and output folders to be used by new instance

The location of these folders can be specified in 3 ways:

- Inside the algorithm – cannot be modified by user
- Algorithm reads 'profile text file' with location of folders
- Location of folders is given as **argument to the algorithm** and as such must be specified in the **argument section of the instance configuration file**

## 7. Create a workflow (\*.xml) which allows executing the new instance:

- Place new workflow xml in:  
*FEWS\FEWS\_SA\Config\WorkflowFiles*
- **In the future: link with export and import routines**

## 8. Register new workflow

- In: *FEWS\FEWS\_SA\Config\RegionConfigFiles\WorkflowDescriptors*  
*1.00 default.xml*

## KEY

- **Design activity**
- **New file needed**
- **Modification to existing file**
- **Activity specific to the GA module**