Delft-FEWS platform implementation at RainGain pilot locations
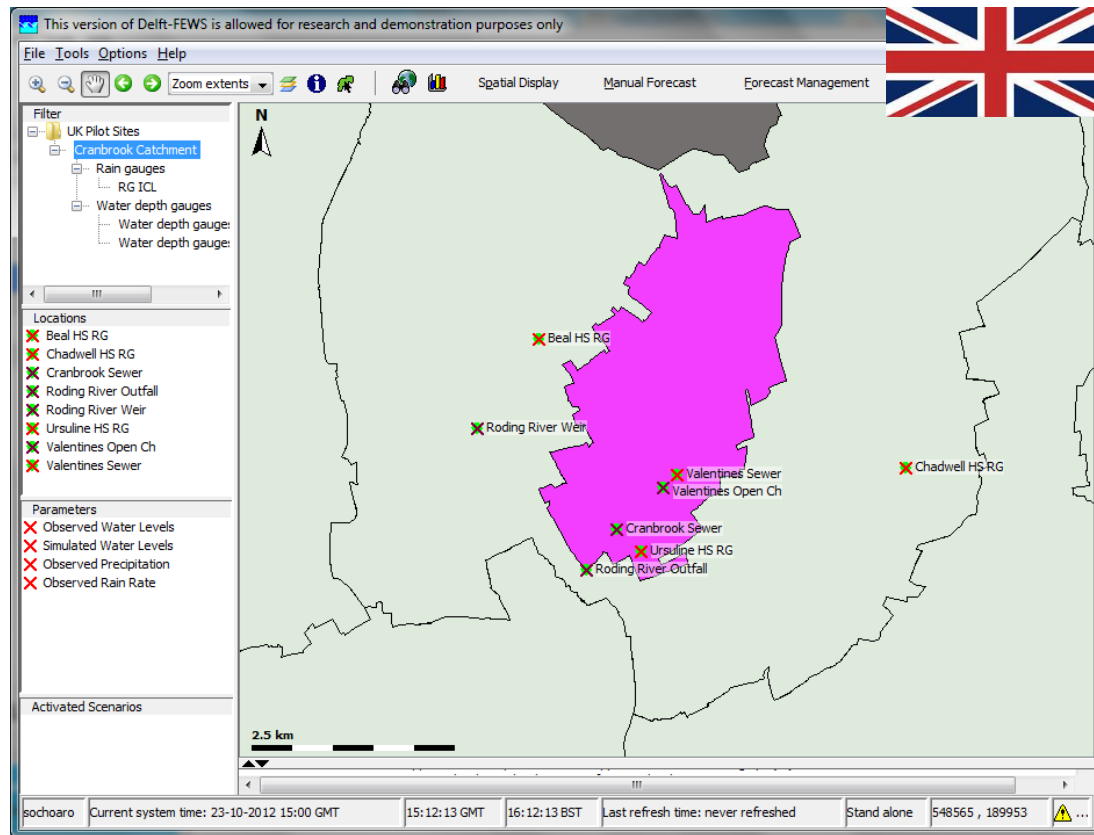
By Susana Ochoa-Rodriguez, Imperial College London
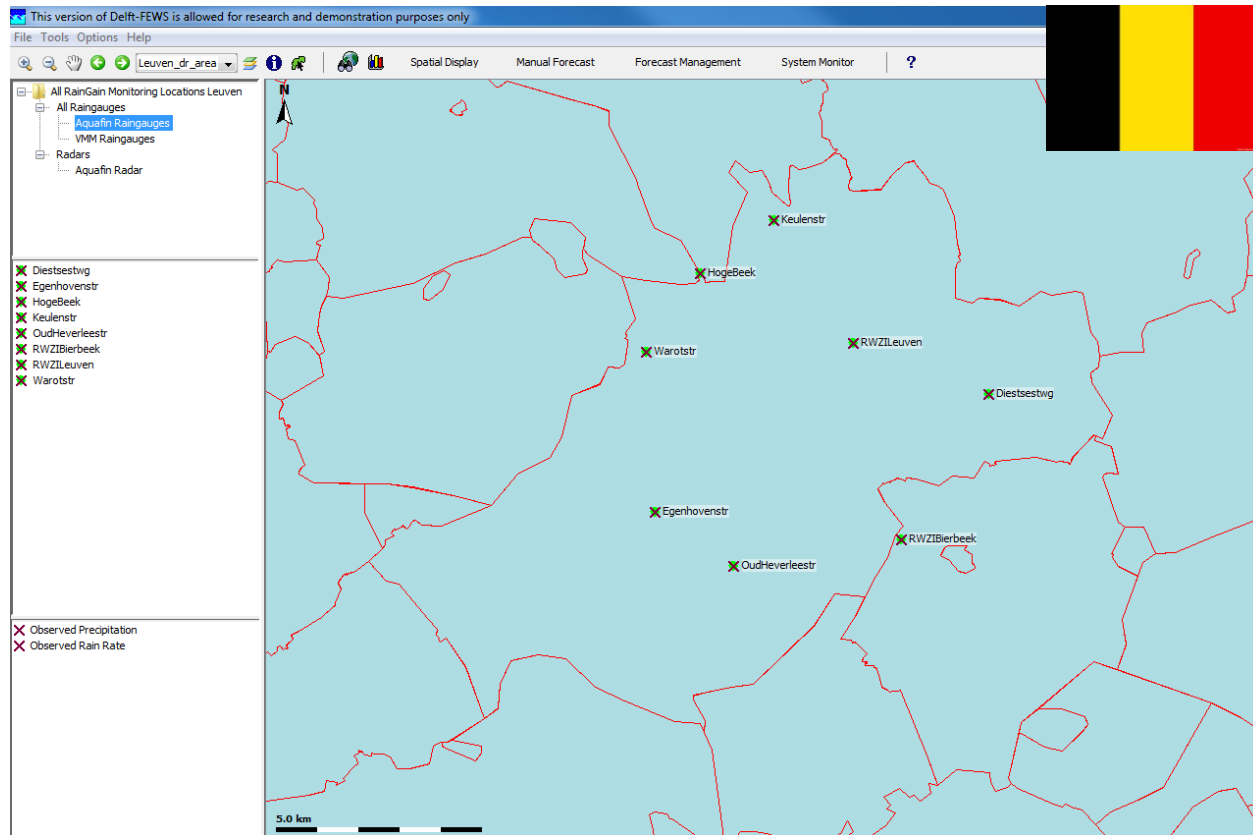(s.ochoa-rodriguez@imperial.ac.uk)

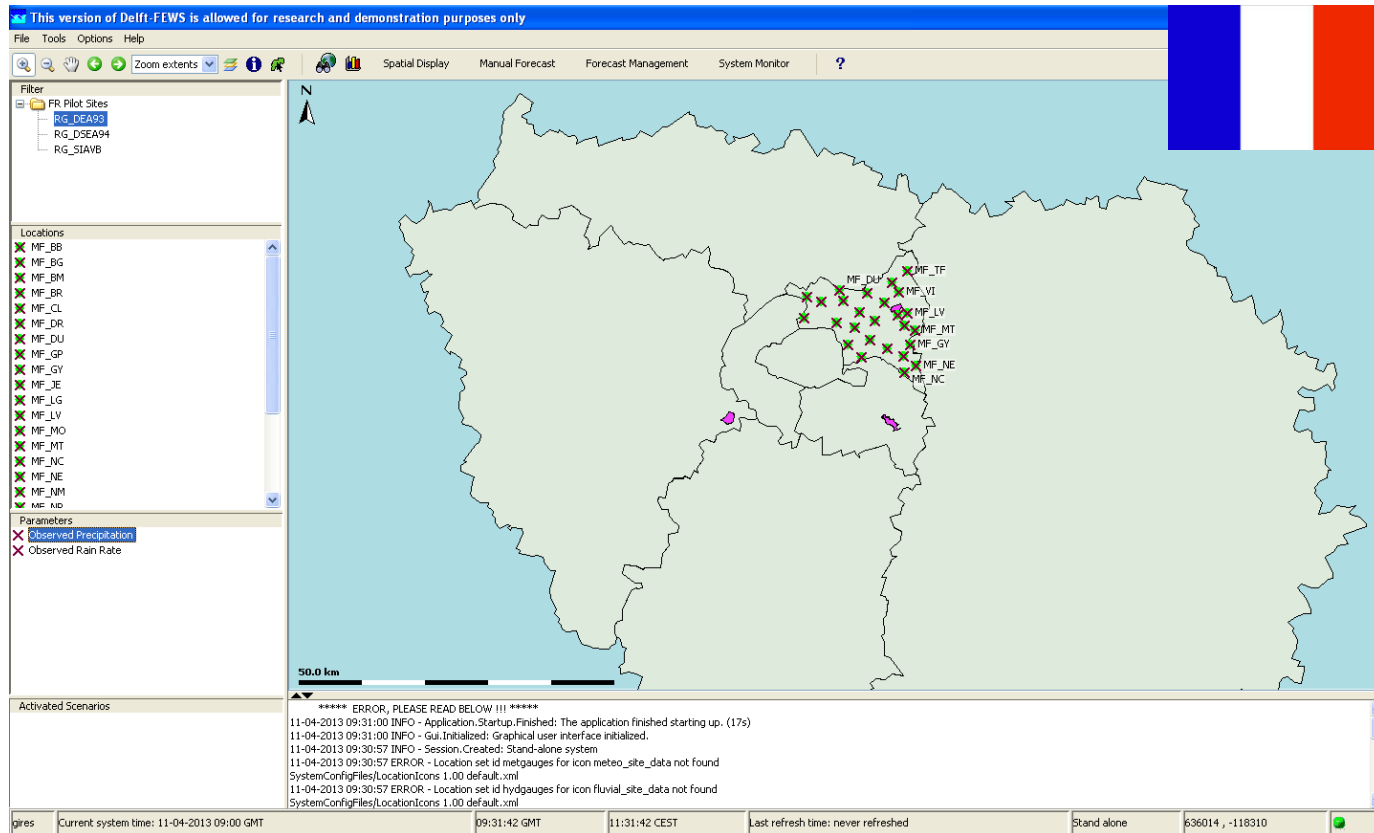*With inputs from coordinators of all of RainGain's pilot locations*

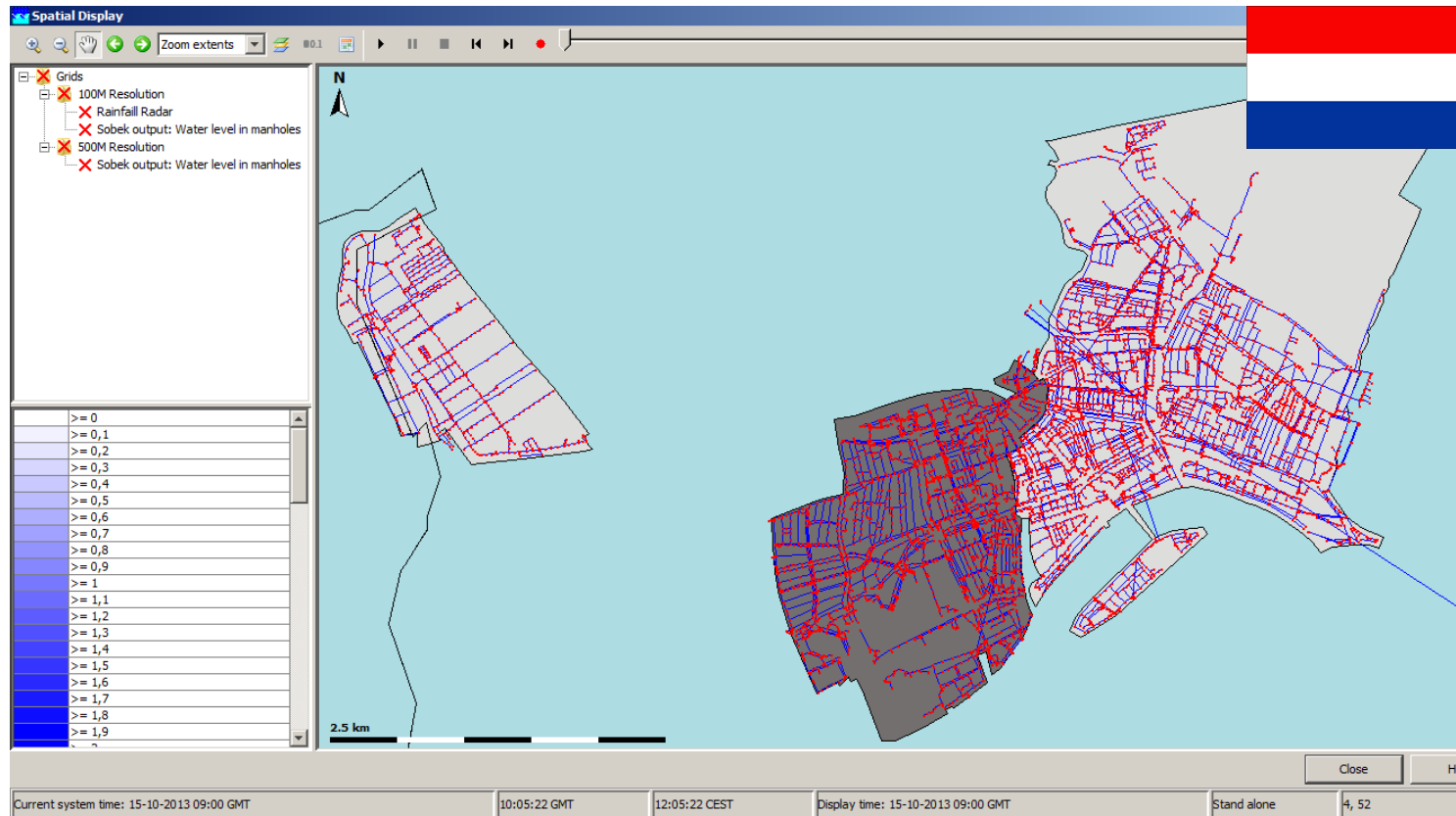# Delft-FEWS platform implementation for UK pilot location

Delft-FEWS platform implementation for Belgian pilot location

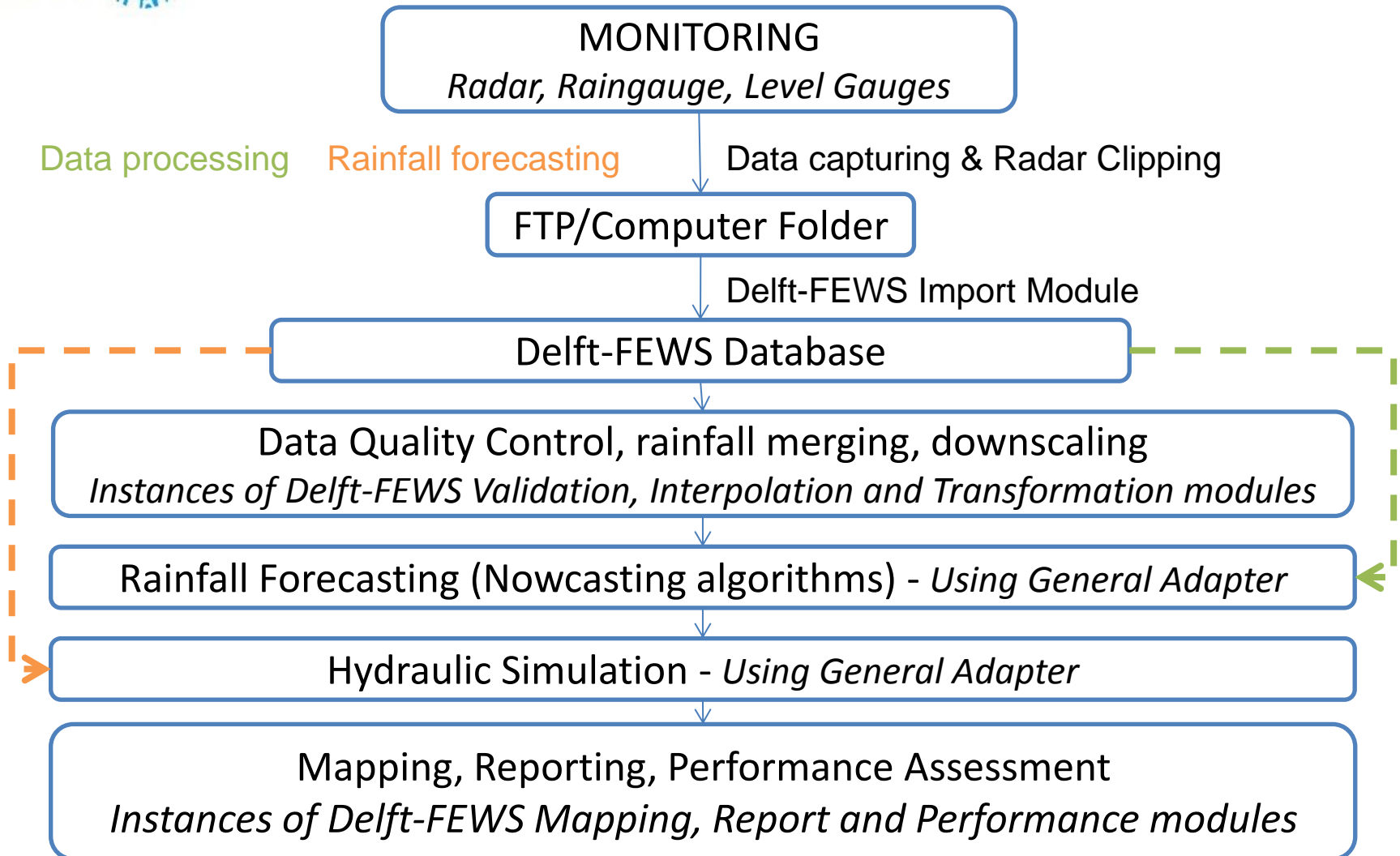# Delft-FEWS platform implementation for French pilot locations

Delft-FEWS platform implementation for Dutch pilot locations

DETAILS OF THE IMPLEMENTATION

- Workflow

- Proposed setup

- Implementation of maps, locations and gauging stations

- Import and visualisation of gauge data

- Connection of hydraulic model and visualisation of simulation results vs. observations

Workflow

MONITORING
Radar, Raingauge, Level Gauges

Data processing   Rainfall forecasting   Data capturing & Radar Clipping

FTP/Computer Folder

Delft-FEWS Import Module

Delft-FEWS Database

Data Quality Control, rainfall merging, downscaling
Instances of Delft-FEWS Validation, Interpolation and Transformation modules

Rainfall Forecasting (Nowcasting algorithms) - *Using General Adapter*

Hydraulic Simulation - *Using General Adapter*

Mapping, Reporting, Performance Assessment
Instances of Delft-FEWS Mapping, Report and Performance modules

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

- Installation of Delft-FEWS

  – Received installation files and tutorial from DeltaresFiles include executable for installation of Java platform, in addition to java classes and regional/application specific configuration files

Local Disk (C:)
  FEWS
    bin
    jre
    FEWS_SA

bin	This folder contains the actual Delft-FEWS software.
jre	This folder contains the Java Runtime Environment
FEWS_SA	In this folder the files for a separate application are stored. There can be multiple application folders next to the bin and jre folder

- Setup of maps, locations and filters:

  - **Maps:** with shapefiles (FEWS_SA\Config\MapLayerFiles), need to customise display (FEWS_SA\Config\SystemConfigFiles\Explorer.xml)

  - **Point locations:** can be defined using shapefiles or specifying coordinates (FEWS_SA\Config\RegionConfigFiles\LocationSets.xml; FEWS_SA\Config\RegionConfigFiles\Locations.xml)

  - **Filters:** allow choosing what you want to display and how you want to group your locations in the display. Some locations may be declared, but through filters you choose if you want to display them or not and you can also assign the parameters that will be associated to each specific location. (FEWS_SA\Config\RegionConfigFiles\Filters.xml)

```xml
<locationSet id="Total_Sensors" name="Total Sensors">
    <esriShapeFile>
        <file>All_Sensors.shp</file> <!-- I loaded the shp conta
        <geoDatum>Ordnance Survey Great Britain 1936</geoDatum>
        <id>%ID_NEW%</id>
        <name>%NAME%</name>
        <description>%DESCRIPT%</description>
        <shortName>%SHORTNAME%</shortName>
        <x>%XCOORD%</x>
        <y>%YCOORD%</y>
        <z>%ZCOORD%</z>
        <attribute id="TYPE">
            <text>%TYPE%</text>
        </attribute>
        <attribute id="SYSTEM">
            <text>%SYSTEM%</text>
        </attribute>
        <attribute id="Cranbrook">
            <text>%Cranbrook%</text>
        </attribute>
        <attribute id="Purley">
            <text>%Purley%</text>
        </attribute>
        <attribute id="Torquay">
            <text>%Torquay%</text>
        </attribute>
    </esriShapeFile>
</locationSet>
```
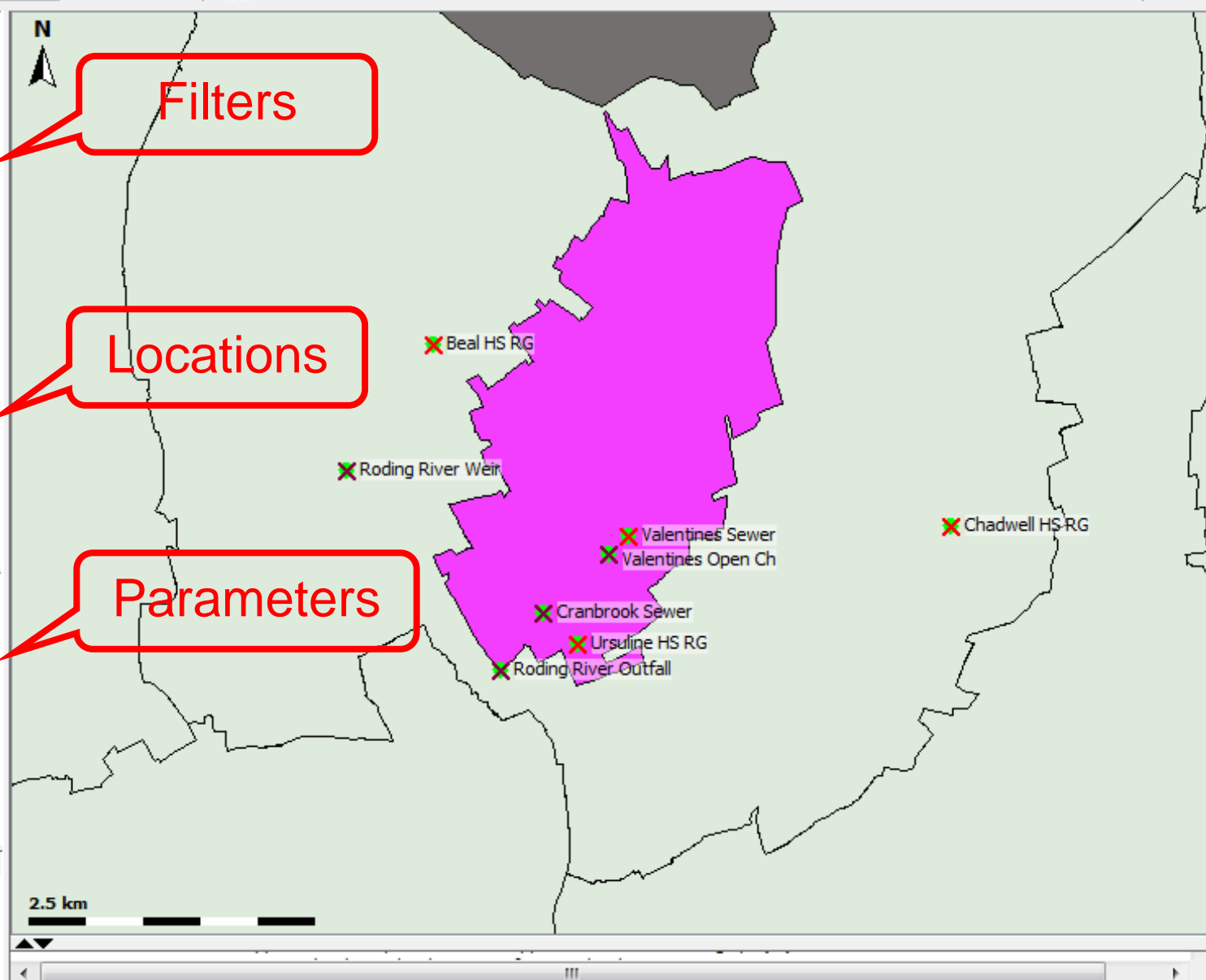
Mapping between shp and FEWS locations

Attributes to be used for filtering later on

Shapefile: Need to think of attributes that will allow filtering in the future

All_Sensors

FID	Shape *	ID	ID_NEW	NAME	DESCRIPT	SHORTNAME	XCOORD	YCOORD	ZCOORD	TYPE	SYSTEM	Cranbrook	Purley	Torquay
0	Point	RG_00	Ascot_Berkshire	Ascot_Berkshire	RG	Ascot_Berkshire	492414.97	167281.853	0	RG	LGfL	0	0	0
1	Point	RG_00	Ashford_Kent	Ashford_Kent	RG	Ashford_Kent	599952.604	141746.819	0	RG	LGfL	0	0	0
2	Point	RG_00	Atomwide_Orpington_S	Atomwide_Orpington_S	RG	Atomwide_Orpington_S	546772.207	167554.175	0	RG	LGfL	0	0	0
3	Point	RG_00	Aylesbury_Buckingham	Aylesbury_Buckingham	RG	Aylesbury_Buckingham	481744.744	211706.987	0	RG	LGfL	0	0	0
4	Point	RG_00	Bedfont_W_London	Bedfont_W London	RG	Bedfont_W_London	509106.765	174286.627	0	RG	LGfL	0	0	0
5	Point	RG_00	Belmont_S_London	Belmont_S London	RG	Belmont_S_London	525318.663	162309.107	0	RG	LGfL	0	1	0
6	Point	RG_00	BETT_Olympia_London	BETT Olympia_London	RG	BETT_Olympia_London	524354.039	178975.595	0	RG	LGfL	0	0	0
7	Point	RG_00	Bodmin_Cornwall	Bodmin_Cornwall	RG	Bodmin_Cornwall	207676.312	65649.348	0	RG	LGfL	0	0	0
8	Point	RG_00	Bow_London	Bow_London	RG	Bow_London	536960.307	182860.526	0	RG	LGfL	0	0	0

- Import and display of telemetry data (raingauge and water depth observations):

  — Using CSV format supported by Delft-FEWS (using existing class)

  — Need to create new instance of existing module
    (FEWS_SA\Config\ModuleConfigFiles)

  — Need to "register" new instance
    (FEWS_SA\Config\RegionConfigFiles\ModuleInstanceDescriptors.xml)

  — Map system variables vs file variables (FEWS_SA\Config\IdMapFiles)

  — Make sure filters will show the imported data
    (FEWS_SA\Config\RegionConfigFiles\Filters.xml)

  — Define workflow (create new one or add activity to existing workflow)
    (FEWS_SA\Config\WorkflowFiles)

  — Register your workflow (FEWS_SA\Config\RegionConfigFiles\WorkflowDescriptors .xml)

CSV format

```
Location Names,Beal_HS_RG,Ursuline_HS_RG,Chadwell_HS_RG
Location Ids,Beal_HS_RG,Ursuline_HS_RG,Chadwell_HS_RG
Time,Rain Rate[mm/h],Rain Rate[mm/h],Rain Rate[mm/h]
2010-08-22 18:00:00,0,0,0
2010-08-22 18:05:00,0,0,0
2010-08-22 18:10:00,0,0,0
2010-08-22 18:15:00,0,0,0
2010-08-22 18:20:00,0,0,0
2010-08-22 18:25:00,0,0,0
2010-08-22 18:30:00,0,0,0
2010-08-22 18:35:00,0,0,0
2010-08-22 18:40:00,0,0,0
2010-08-22 18:45:00,0,0,0
2010-08-22 18:50:00,0,0,0
2010-08-22 18:55:00,0,0,0
2010-08-22 19:00:00,0,0,0
2010-08-22 19:05:00,0,0,0
2010-08-22 19:10:00,0,0,0
2010-08-22 19:15:00,0,0,0
2010-08-22 19:20:00,0,0,0
2010-08-22 19:25:00,0,0,0
2010-08-22 19:30:00,0,0,0
2010-08-22 19:35:00,0,0,0
2010-08-22 19:40:00,0,0,0
2010-08-22 19:45:00,0,2.4,0
2010-08-22 19:50:00,2.4,0,0
2010-08-22 19:55:00,0,0,2.4
2010-08-22 20:00:00,0,2.4,0
2010-08-22 20:05:00,2.4,0,0
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!--NFFS Midlands region-->
<timeSeriesImportRun xmlns="http://www.wldelft.nl/fews" xmlns:xsi="http://www.w3.org/
  <import>
  <general>
      <importType>CSV</importType>
      <folder>$IMPORT_CSV_FOLDER$</folder>
      <failedFolder>$IMPORT_FAILED_FOLDER$</failedFolder>
      <backupFolder>$BACKUP_CSV_FOLDER$</backupFolder>
      <idMapId>IdImportCSV</idMapId>
      <unitConversionsId>ImportUnitConversions</unitConversionsId>
      <importTimeZone>
        <timeZoneOffset>-23:00</timeZoneOffset>
      </importTimeZone>
      <dataFeedId>CSV</dataFeedId>
   </general>
   <timeSeriesSet>
     <moduleInstanceId>ImportCSV</moduleInstanceId>
     <valueType>scalar</valueType>
     <parameterId>RR.obs</parameterId>
     <locationSetId>RG_ICL_Cranbrook</locationSetId>
     <timeSeriesType>external historical</timeSeriesType>
     <timeStep unit="minute" multiplier="5"/> <!--Temporal resolution-->
     <readWriteMode>read only</readWriteMode>
     <synchLevel>1</synchLevel>
   </timeSeriesSet>
   <timeSeriesSet>
     <moduleInstanceId>ImportCSV</moduleInstanceId>
     <valueType>scalar</valueType>
```

New module instance for importing CSV files

IdMapping for new CSV import instance

```xml
<?xml version="1.0" encoding="UTF-8"?>
<idMap xmlns="http://www.wldelft.nl/fews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocat
   <!-- these are the parameters -->
   <parameter internal="P.obs" external="Rainfall"/> <!--External is the name to be included in the CSV file-->
   <parameter internal="RR.obs" external="Rain Rate"/>
   <parameter internal="H.obs" external="Water Depth"/>
   <!-- these are the locations - So far I have only included the locations of Cranbrook ICL sensors-->
   <location internal="Beal_HS_RG" external="Beal_HS_RG"/>
   <location internal="Ursuline_HS_RG" external="Ursuline_HS_RG"/>
   <location internal="Chadwell_HS_RG" external="Chadwell_HS_RG"/>
   <location internal="Valentines_Sewer" external="Valentines_Sewer"/>
   <location internal="Valentines_OpenC" external="Valentines_OpenC"/>
   <location internal="Cranbrook_Sewer" external="Cranbrook_Sewer"/>
   <location internal="Roding_Outfall" external="Roding_Outfall"/>
   <location internal="Roding_River" external="Roding_River"/>
 </idMap>
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<workflow xmlns="http://www.wldelft.nl/fews" xmlns:xsi="http://www.w3.org/2001/XML

  <!--Import CSV Time Series from Telemetry-->
  <activity>
    <runIndependent>true</runIndependent>
    <moduleInstanceId>ImportCSV</moduleInstanceId>
  </activity>
</workflow>
```
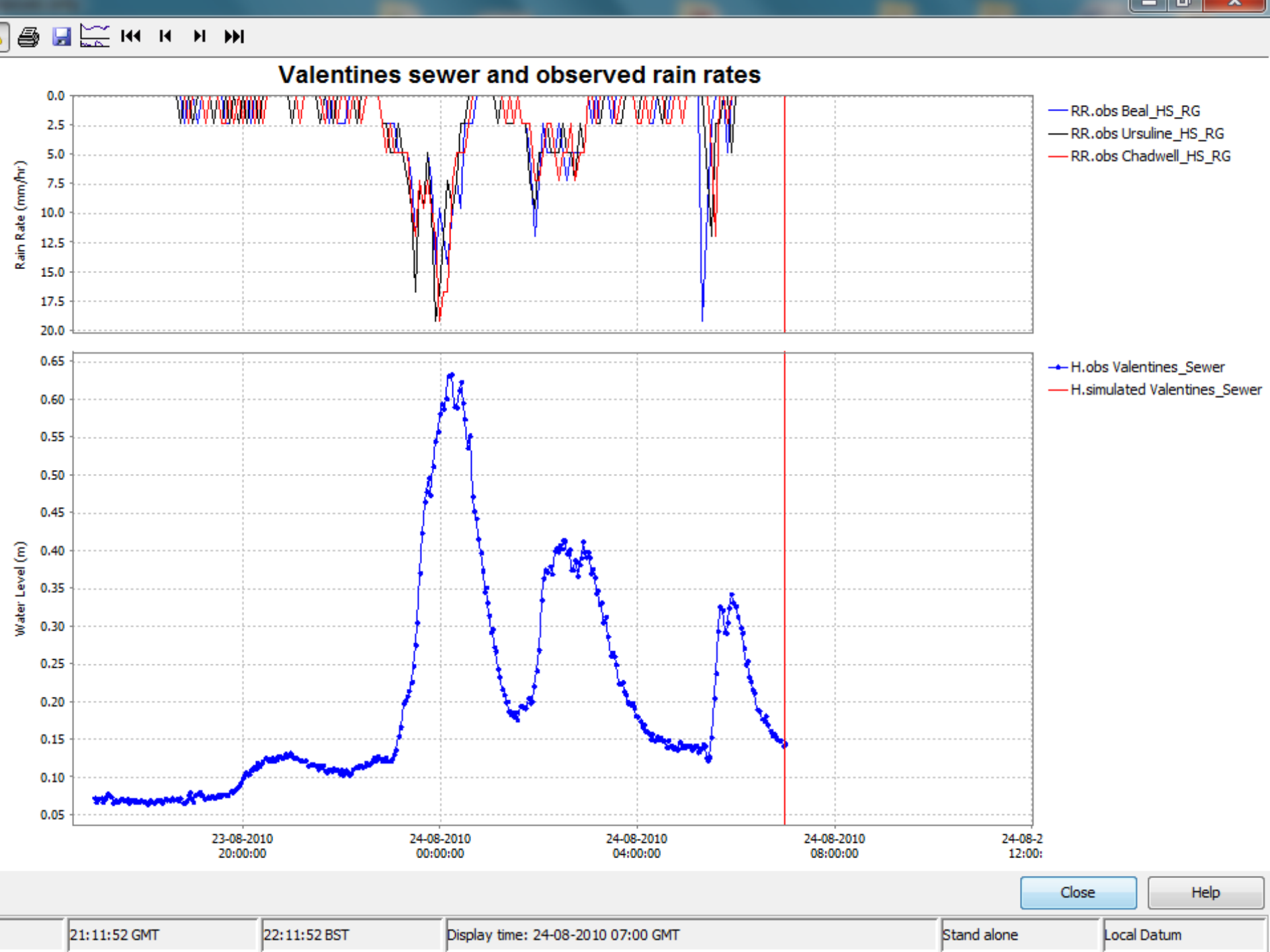
New workflow

Registration of new workflow

```xml
<?xml version="1.0" encoding="UTF-8"?>
<workflowDescriptors xmlns="http://www.wldelft.nl/fews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-i
  <workflowDescriptor id="ImportExternal" visible="true" forecast="false">
    <description>Import external data</description>
  </workflowDescriptor>
  <workflowDescriptor id="RunSwmm" visible="true" forecast="false">
    <description>Run SWMM</description>
  </workflowDescriptor>
  <workflowDescriptor id="ImportCSVTelemetry" visible="true" forecast="false">
    <description>Import rainfall telemetry</description>
  </workflowDescriptor>
</workflowDescriptors>
```
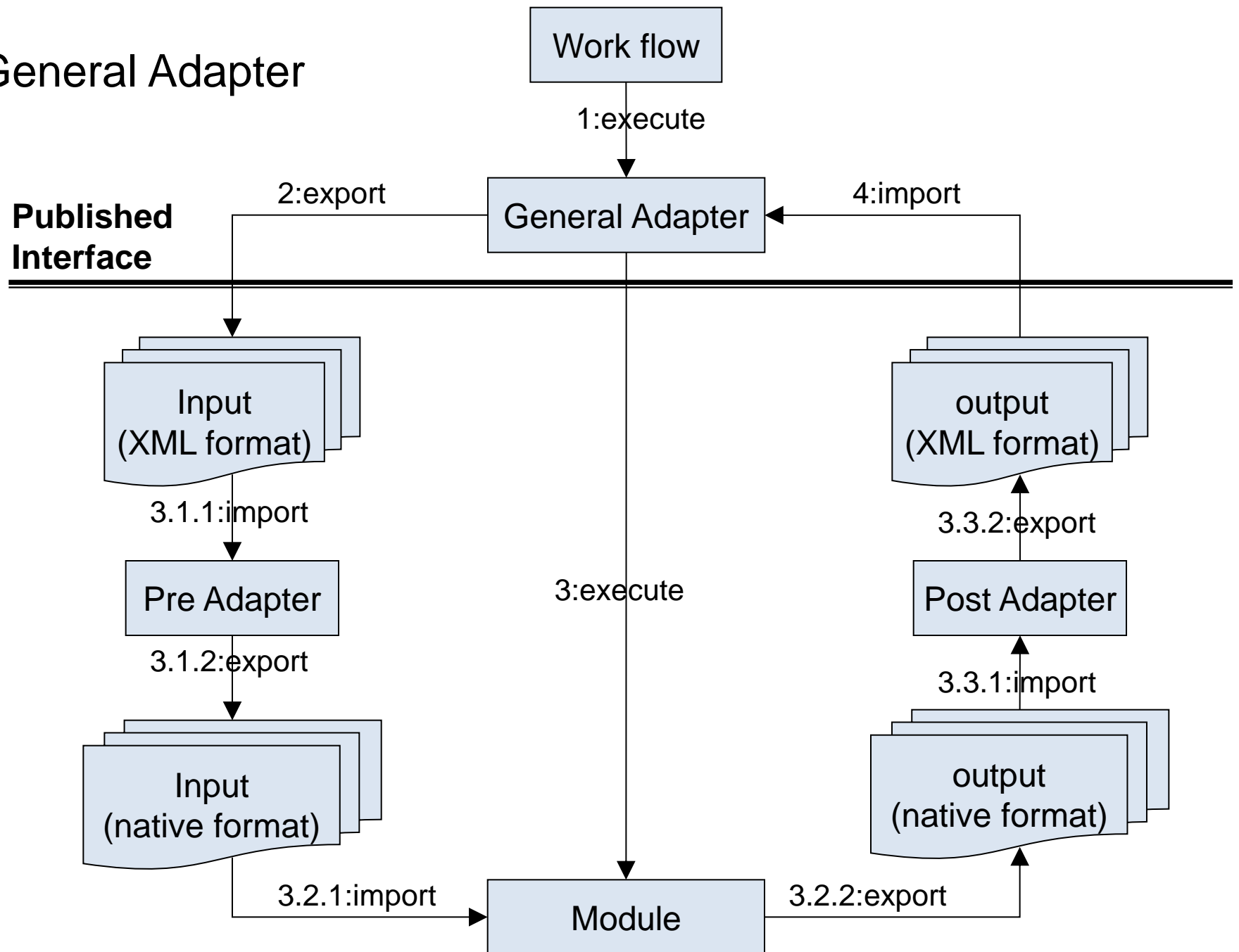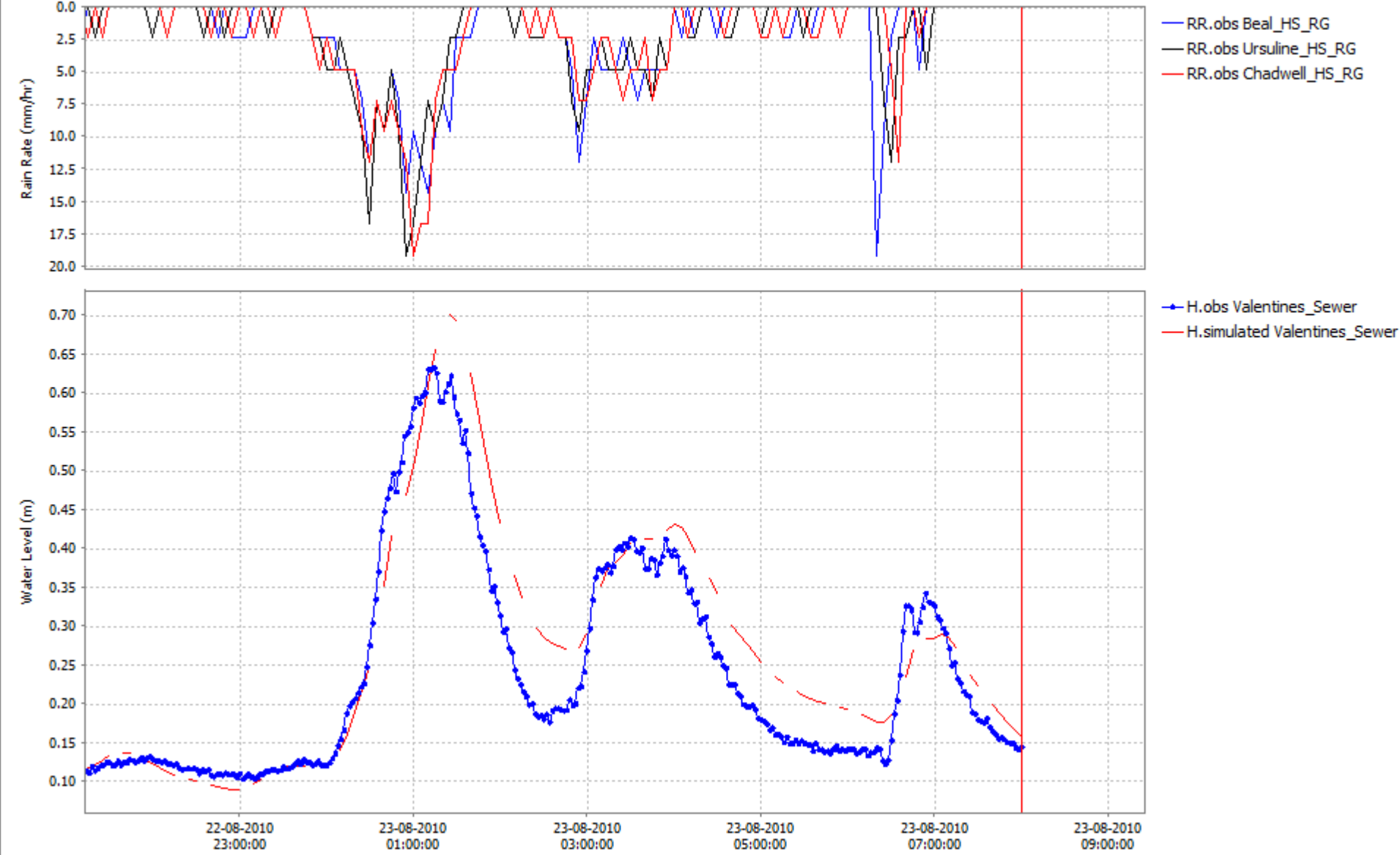
Valentines sewer and observed rain rates

- Linkage and running of hydraulic model (SWMM)

  — Used computing engine (.exe) of SWMM

  — Used "General Adapter" – created new instance of "General Adapter" called "swmm", which "calls" the computing engine of SWMM and runs it.

  — The detailed activities required for linking and running SWMM are the following:

    ▪ *Pre-adapter:* Transformation of data into SWMM format

    ▪ *Main instance:* Running of SWMM   (FEWS_SA\Modules\swmm)
      (FEWS_SA\ModuleConfigFiles\swmm)

    ▪ *Post-adapter:* Parsing of SWMM output and converting to CSV (an .exe was created for this) (FEWS_SA\ModuleConfigFiles\swmm_output)

    ▪ Importing CSV output back to SWMM (using same module instance used for importing telemetry data)

  — The new instances must be registered and included in a workflow, as was done with the CSV import instance.

General Adapter

Work flow

1:execute

2:export   General Adapter   4:import

Published Interface

Input (XML format)

3.1.1:import

Pre Adapter

3.1.2:export

Input (native format)

3.2.1:import   Module   3.2.2:export

3:execute

output (XML format)

3.3.2:export

Post Adapter

3.3.1:import

output (native format)

```xml
<?xml version="1.0" encoding="UTF-8"?>
<workflow xmlns="http://www.wldelft.nl/fews" xmlns:xsi="http://www.w3.org/2001
  <!--Interpolate Hydro1 and Hydro2        -->
  <activity>
    <runIndependent>true</runIndependent>
    <moduleInstanceId>ImportCSV</moduleInstanceId>
  </activity>
  <activity>
    <runIndependent>true</runIndependent>
    <moduleInstanceId>ExportCSV</moduleInstanceId>
  </activity>
  <activity>
    <runIndependent>true</runIndependent>
    <moduleInstanceId>swmm</moduleInstanceId>
  </activity>
  <activity>
    <runIndependent>false</runIndependent>
    <moduleInstanceId>swmm_output</moduleInstanceId>
  </activity>
  <activity>
    <runIndependent>false</runIndependent>
    <moduleInstanceId>ImportSwmmCSV</moduleInstanceId>
  </activity>
</workflow>
```

Valentines sewer and observed rain rates

Legend (top chart):
- RR.obs Beal_HS_RG
- RR.obs Ursuline_HS_RG
- RR.obs Chadwell_HS_RG

Legend (bottom chart):
- H.obs Valentines_Sewer
- H.simulated Valentines_Sewer

In the same way we ran SWMM, any other *.exe programme can be linked to Delft-FEWS in the future

- Import and visualisation of radar data (grid display):

  — Using Import Nimrod routines supported by Delft-FEWS (using existing class)

  — Need to create new instance of existing module
  (FEWS_SA\Config\ModuleConfigFiles)

  — Need to "register" new instance
  (FEWS_SA\Config\RegionConfigFiles\ModuleInstanceDescriptors.xml)

  — Map system variables vs file variables (FEWS_SA\Config\IdMapFiles)

  — Configure grid display
  (FEWS_SA\Config\DisplayConfigFiles\SpatialDisplay 1.00 Default.xml)

  — Define workflow (create new one or add activity to existing workflow)
  (FEWS_SA\Config\WorkflowFiles)

  — Register your workflow (FEWS_SA\Config\RegionConfigFiles\WorkflowDescriptors .xml)