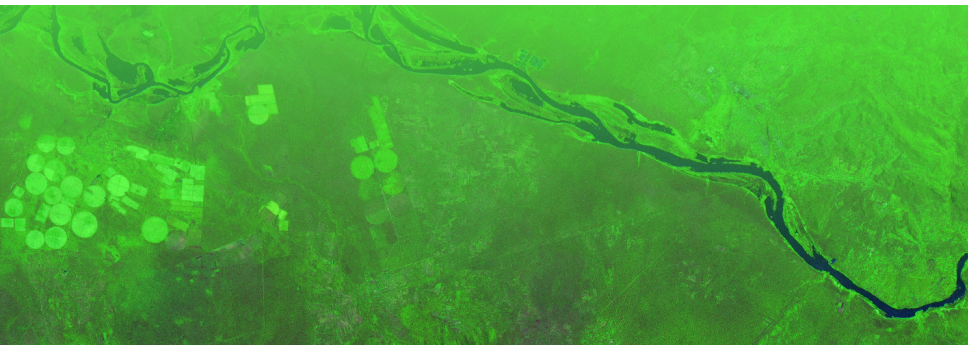


Linking water resource network models to an open data management platform

Philipp Meier (philipp.meier@eawag.ch)

University of Manchester – School of Mechanical, Aerospace and Civil Engineering, UK
EAWAG: Swiss Federal Institute of Aquatic Science and Technology, Switzerland

iEMSs 2014, June 14 – 18, San Diego, CA



Water Resources Modelling

Most models used in water resources modelling don't feature a user interface.

- Huge diversity of models
- Model input is stored in files
 - Building models is error prone and time consuming.
 - Requires file management in addition to data management.
 - It is difficult to run different models on the same dataset.

Water Resources Modelling

Most models used in water resources modelling don't feature a user interface.

- Huge diversity of models
- Model input is stored in files
 - Building models is error prone and time consuming.
 - Requires file management in addition to data management.
 - It is difficult to run different models on the same dataset.

Water Resources Modelling

Most models used in water resources modelling don't feature a user interface.

- Huge diversity of models
- Model input is stored in files
 - Building models is error prone and time consuming.
 - Requires file management in addition to data management.
 - It is difficult to run different models on the same dataset.

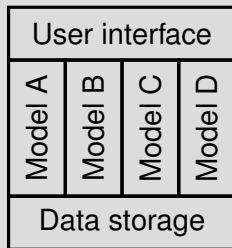
Water Resources Modelling

Most models used in water resources modelling don't feature a user interface.

- Huge diversity of models
- Model input is stored in files
 - Building models is error prone and time consuming.
 - Requires file management in addition to data management.
 - It is difficult to run different models on the same dataset.

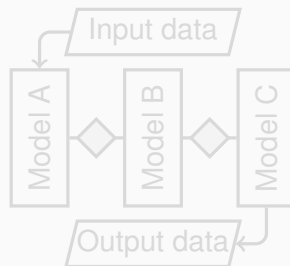
Modelling software architectures

Modelling shell



- finite set of models
- close integration of user interface and data storage

Modelling framework



- Inter-model communication through standardised interface

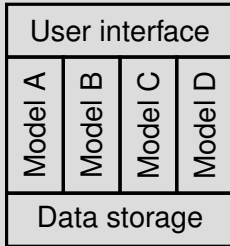
Model platform



- Independent components
- Common data storage

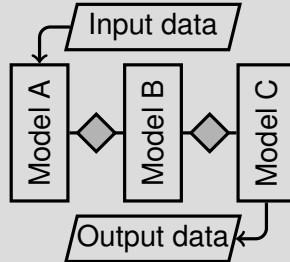
Modelling software architectures

Modelling shell



- finite set of models
- close integration of user interface and data storage

Modelling framework



- Inter-model communication through standardised interface

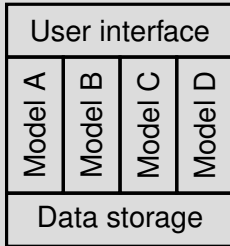
Model platform



- Independent components
- Common data storage

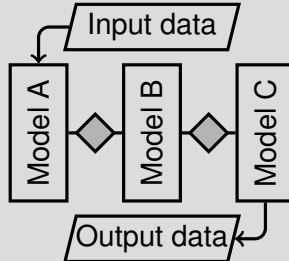
Modelling software architectures

Modelling shell



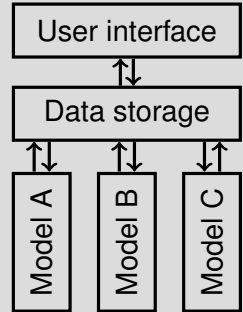
- finite set of models
- close integration of user interface and data storage

Modelling framework



- Inter-model communication through standardised interface

Model platform



- Independent components
- Common data storage

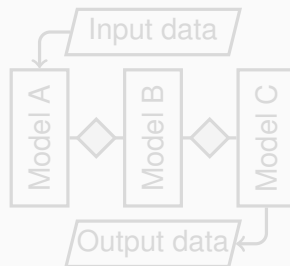
Modelling software architectures

Modelling shell



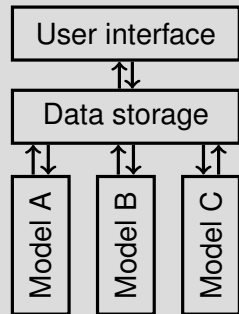
- finite set of models
- close integration of user interface and data storage

Modelling framework



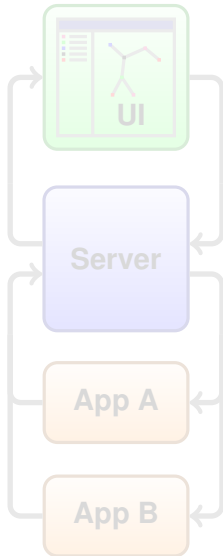
- Inter-model communication through standardised interface

Model platform



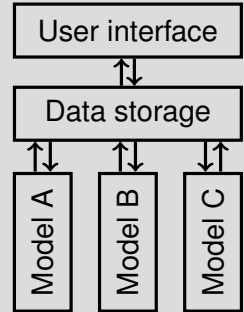
- Independent components
- Common data storage

Hydra Platform



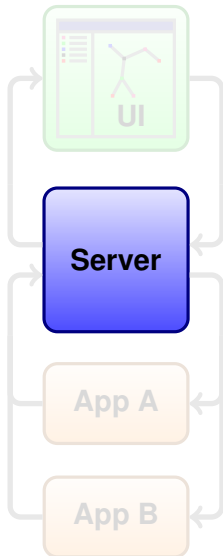
- State of the art UI using MapWindow libraries
- Server can be installed locally or remotely
- Apps connect models to the data in the server

Model platform



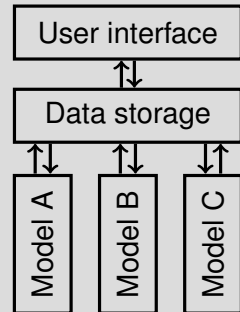
- Independent components
- Common data storage

Hydra Platform



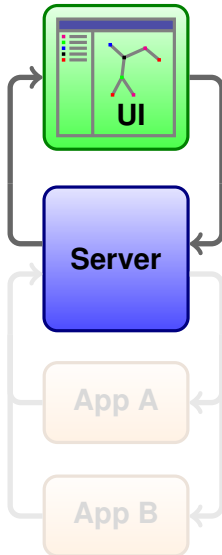
- State of the art UI using MapWindow libraries
- Server can be installed locally or remotely
- Apps connect models to the data in the server

Model platform



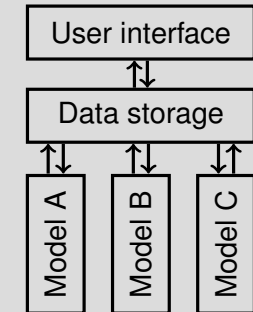
- Independent components
- Common data storage

Hydra Platform



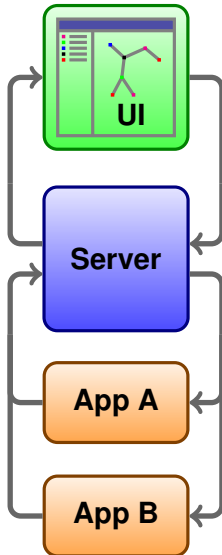
- State of the art UI using MapWindow libraries
- Server can be installed locally or remotely
- Apps connect models to the data in the server

Model platform



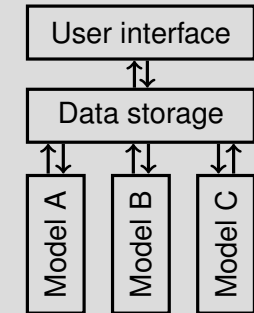
- Independent components
- Common data storage

Hydra Platform



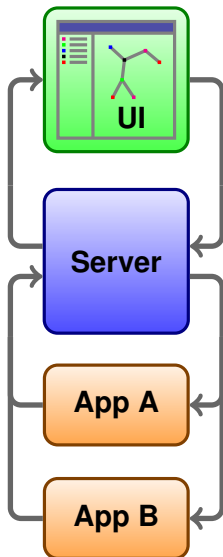
- State of the art UI using MapWindow libraries
- Server can be installed locally or remotely
- Apps connect models to the data in the server

Model platform

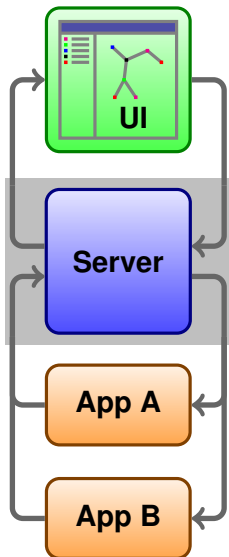


- Independent components
- Common data storage

Server functionality



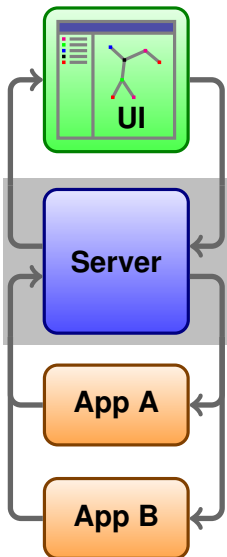
Server functionality



Database

- Stores networks consisting of nodes and links and attributes
- Designed for maximum compatibility with heterogeneous data models

Server functionality

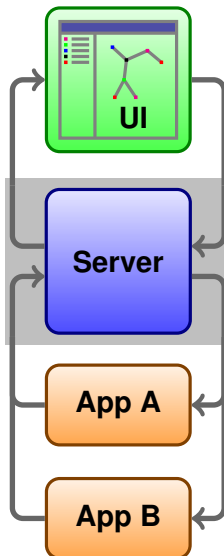


Database

Data
management

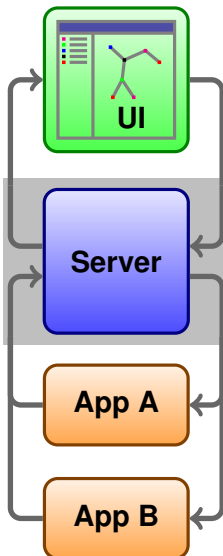
- Stores networks consisting of nodes and links and attributes
- Designed for maximum compatibility with heterogeneous data models
- Data management functionality (unit conversion, time zone handling, consistency checks, ...)

Server functionality

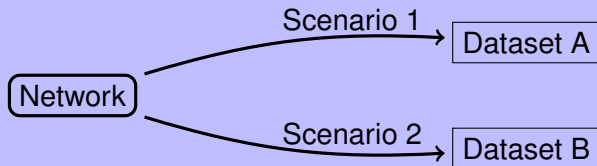


- Stores networks consisting of nodes and links and attributes
- Designed for maximum compatibility with heterogeneous data models
- Data management functionality (unit conversion, time zone handling, consistency checks, ...)
- Interface for data exchange between Apps and the database
 - Based on SOAP (web service)
 - High level access to data stored in the database

Networks and scenarios

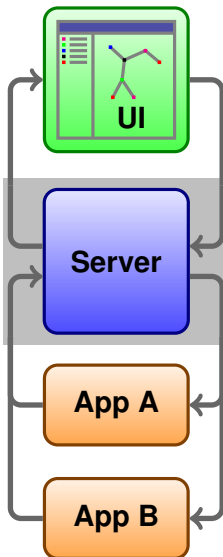


- Network topology and associated data are stored independently

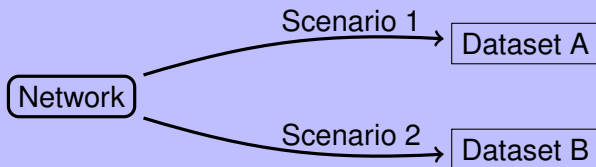


- A network stores the complete topology and attributes associated with each element (node, link).
- A scenario assigns data to the different attributes.

Networks and scenarios

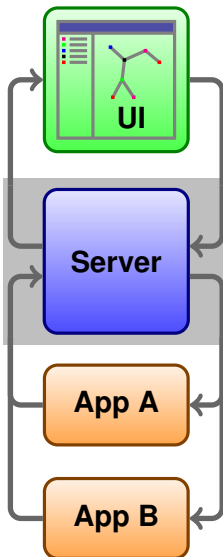


- Network topology and associated data are stored independently



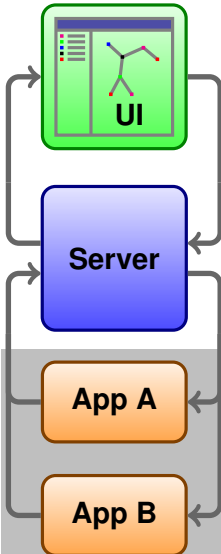
- Storage of different data types
 - scalars
 - strings
 - time series data
 - multi-dimensional arrays

Server implementation



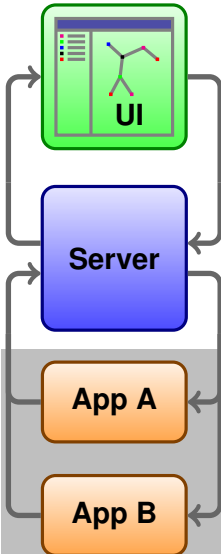
- Open source
 - Will be released under GPL licence
- Written in Python
 - Cross-platform
- Different installation options
 - Local install
 - Install on server behind an Apache web server (WSGI)
- Multi-threaded

Apps



- Standalone software
- Communicate with server API
- Implemented in any programming language that supports web services (SOAP):
 - Python
 - MATLAB
 - Java
 - VBA
 - VB .NET
 - ...
- Developer libraries are provided for Python and C#

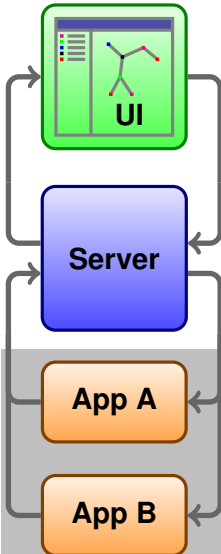
Apps



Different types of Apps:

- Apps for data manipulation
 - data aggregation / disaggregation
 - ...
- Data import / export
 - Import data from and export to various data formats (CSV, WaterML, HEC-DSS, ...)
- Model interfaces
 - Software that runs model using data retrieved from the server
 - Wrapper around model

Apps

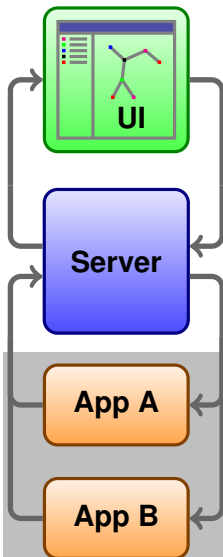


Different types of Apps:

- Apps for data manipulation
 - data aggregation / disaggregation
 - ...
- Data import / export
 - Import data from and export to various data formats (CSV, WaterML, HEC-DSS, ...)
- Model interfaces
 - Software that runs model using data retrieved from the server
 - Wrapper around model

! Each model has its own requirements for node and link attributes

Apps – Templates

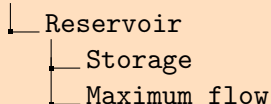


! Each model has its own requirements for node and link attributes

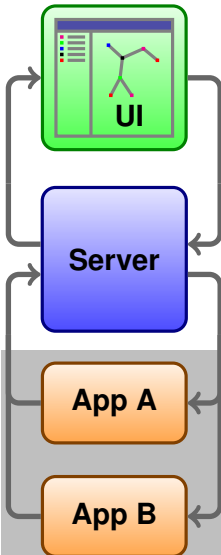
Template

- A template provides the building blocks of a network model
- Defines the set of attributes for different node or link types used for one specific model

Water Resources Model



Apps – Templates



! Each model has its own requirements for node and link attributes

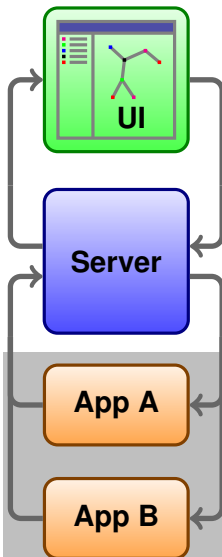
Template

```

<template_definition>
<template_name>WR Model</template_name>
<resources>
<resource>
  <type>NODE</type>
  <name>Reservoir</name>
  <attribute>
    <name>Storage</name>
    <dimension>Volume</dimension>
    <is_var>Y</is_var>
  </attribute>
  ...

```

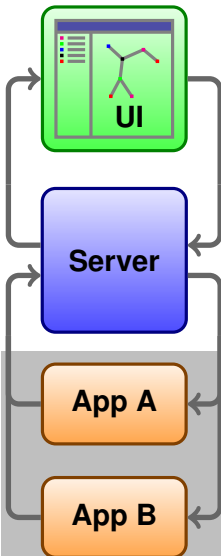
Apps – Templates



! Each model has its own requirements for node and link attributes

→ A single node (or link) can satisfy more than one template

Apps – Templates



! Each model has its own requirements for node and link attributes

→ A single node (or link) can satisfy more than one template

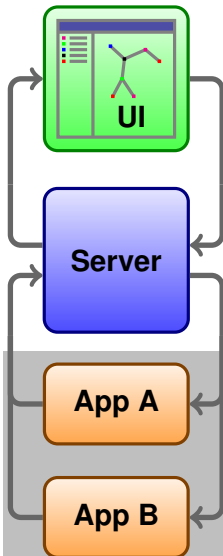
Reservoir Model A

- Storage
- Max level
- Min level
- Level - area
- volume

Reservoir Model B

- Storage
- Spillgate release
- Irrigation demand

Apps – Library



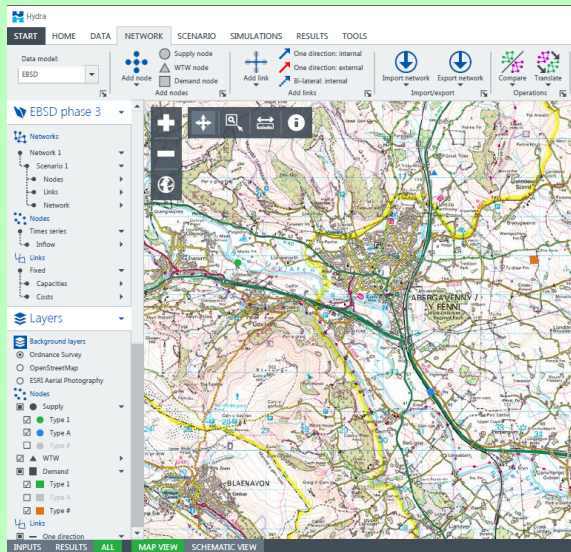
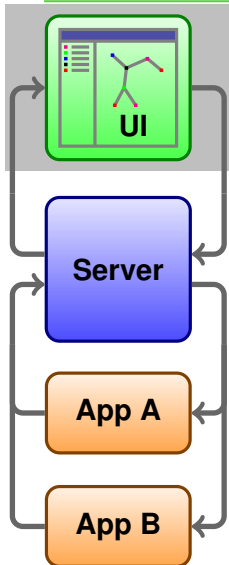
→ A single node (or link) can satisfy more than one template

Library

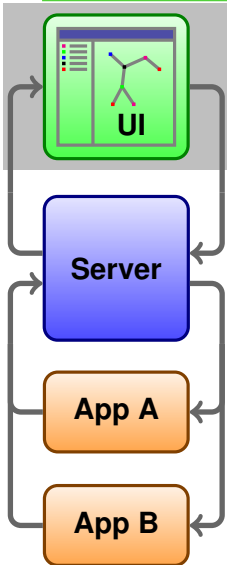
- A library is controlled vocabulary describing the types and attributes of nodes and links.
- A library acts as a guideline for the creation of templates
- A library is agreed on by the research community

Data exchange between Apps that adhere to the same library is guaranteed to work.

User interface

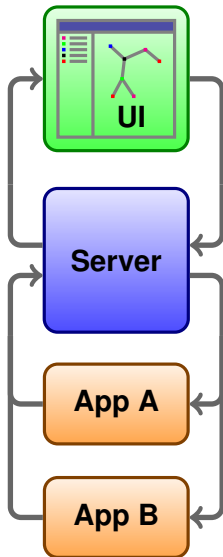


User interface



- State of the art Windows desktop application
- Interact with the Hydra server in an intuitive way
 - Create network visually
 - Build scenarios
 - Edit data
 - Run App
 - Visualise model results
- Developed by CH2M Hill

Summary



Hydra Platform is a data management tool for network based resources models following a distributed software architecture.

- Open source server as core application
 - Consistent database for network model data
 - Feature-rich interface for Apps
 - Cross-platform
- Apps for data import / export and connectors to models
- Templates and Libraries facilitate data exchange between models.
- State-of-the art user interface



Thank you!

More about Hydra Platform:

Talk by S. Knox:

“Web service and plug-in architecture for flexibility and openness of environmental data sharing platforms”

Session A1 at 10:40

A minimal App in Python

```
from HydraLib import PluginLib  
  
client = PluginLib.connect()  
network = client.service.get_network(2)
```

Glossary

Data model: “[...] is a description of the objects represented by a computer system together with their properties and relationships; these are typically ‘real world’ objects such as products, suppliers, customers, and orders.” (Wikipedia)

Data schema: “A data schema of a database system is its structure described in a formal language supported by the database management system (DBMS) and refers to the organization of data as a blueprint of how a database is constructed [...]” (Wikipedia)