

# Deltares

## Developing Smart Water Systems in Open Source with RTC-Tools

Jesús A. Rodríguez

AIWW 2023

7/11/2023

# Workshop outline

- Smart water systems
- RTC-Tools overview
- Case studies
- Practical session

# Smart Water Systems

# What are Smart Water Systems?

Smart water systems incorporate advanced control technologies and IT technologies to water infrastructure and consumer usage. The primary goal is to improve real-time decisions across the system.

# What are the components of Smart Water Systems?

- Data collection: Sensors and meters
- Data transmission: Communication networks
- Data processing: Data analytics tools
- Data visualization and interaction: user interfaces
- Decision-making: Optimization, simulation and automated control technologies

# What are the potential benefits of Smart Water Systems

- Improved water efficiency
- Enhanced system reliability
- Increased customer engagement
- Reduced operational costs
- Greater environmental sustainability

# Application case: Western Municipal Water District (WMWD) of California utilities [1]

- SCADA system to manage real-time alarms and automatically operate plants and networks
- Benefits: 30% savings on energy use, a 20% decrease in water loss, and a 20% decline in disruption

# Group activity

- What are the key challenges in implementing smart water systems in your organization, and how can these challenges be addressed?
- What are some potential benefits of smart water systems in your organization, and how those benefits could be achieved?

# RTC-Tools overview

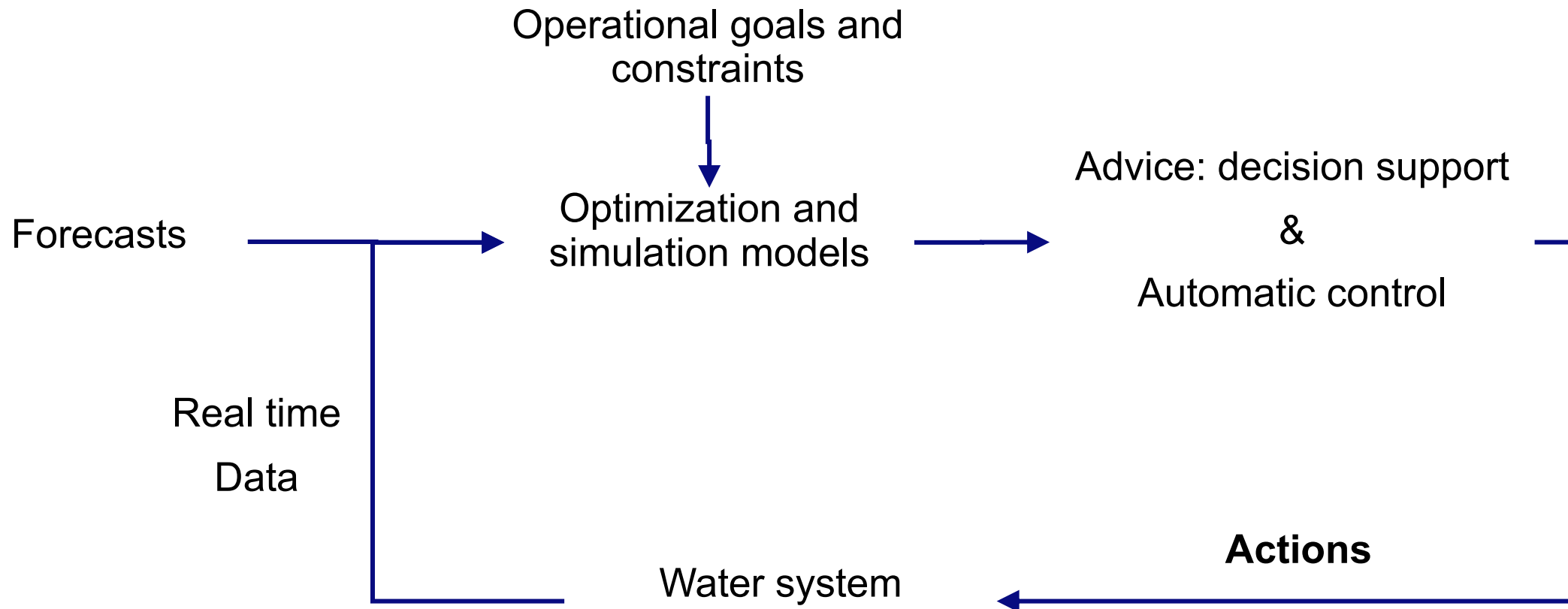
# Challenges in Water System Operation

- Changing hydrological conditions with extremes
- Uncertainty in hydrological forecasts/demand
- Seasonal variation: hydrological conditions and operational objectives change in summer and winter
- Inter-dependent decisions in space and time
- Conflicting operational objectives from different stakeholders;
- Changing demographics
- linear and nonlinear process equations with continuous and discrete elements
- Energy costs



Operational goals and constraints

# From data to actions

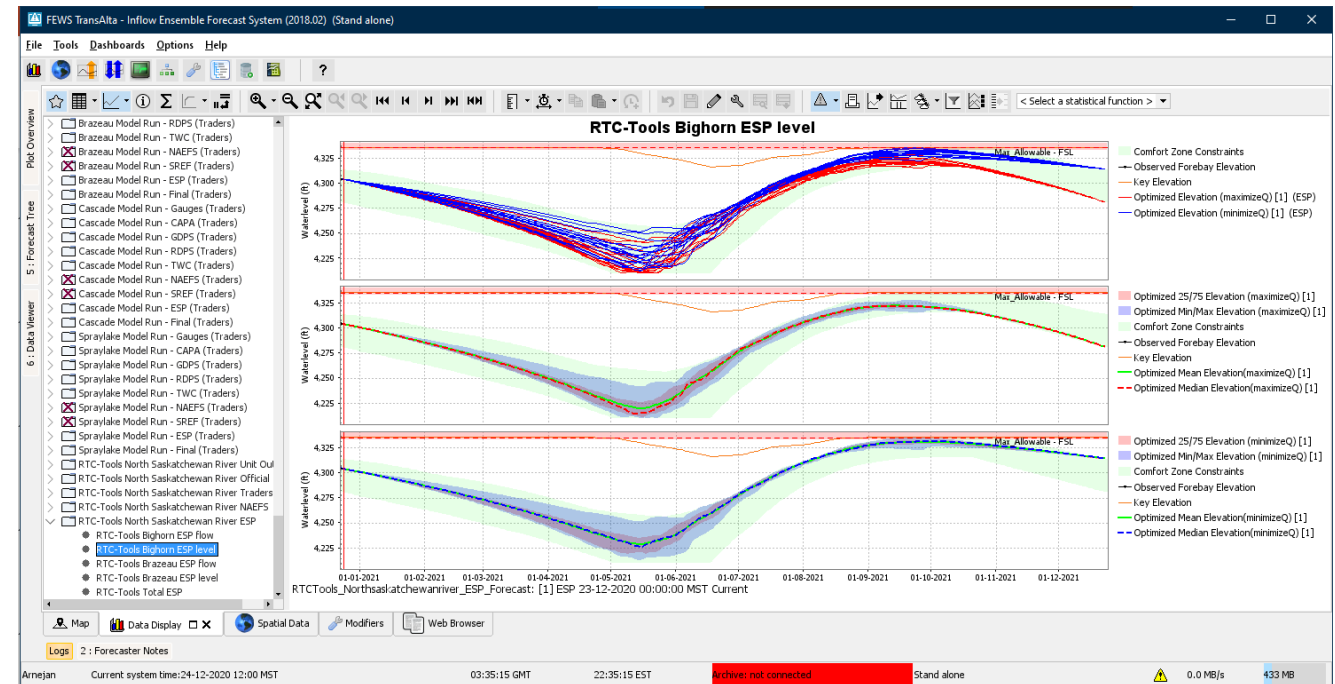


# RTC-Tools



A toolbox for (Real-Time) Control, simulation and optimization

- Open source
- Python package
- Expandible modeling components
- Optimization under multiple goals and uncertainty
- Support for user interface integration



RTC-Tools optimization results in Delft-FEWS TransAlta (van Loenen & Fru, CEATI conference 2021)

# RTC-Tools as open software

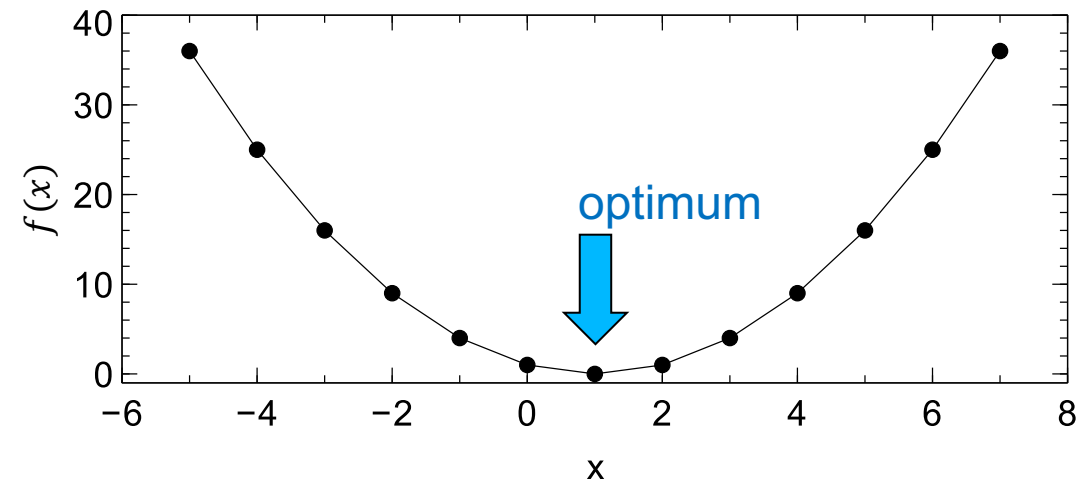
- Open-source code in Python
- Open modeling libraries (Modelica)
- Supports open-source solvers: CBC, HiGHS, Ipopt
- Uses CasADi (Open source) as symbolic framework
- Seamless integration with Delft-FEWS

# RTC-Tools supports mathematical optimization

**Mathematical optimization** is the selection of a best element,  $x^*$ , (with regard to some criterion) from a set of available alternatives.

$$\begin{array}{ll} \min f(x) & \text{subject to} \\ g(x) \leq 0 & \\ h(x) = 0 & \end{array}$$

- Minimize  $f$  (objective function)
  - Contains deviation from a setpoint, maximize power, other expressions
- Inequality constraints  $g$ 
  - Used for physical limits like min, max of water level, volume, turbine flow
- Equality constraints  $h$ 
  - Used for flow equations, water balance
- Find optimal solution  $x^*$

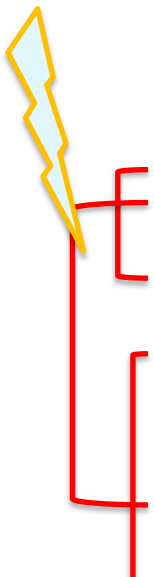


# RTC-Tools modeling functionalities

- Simulation
  - Discrete time
- Optimization
  - Linear
  - Non-linear
  - Mixed-integer: logic decisions
  - Stochastic: ensemble forecasts
  - Goal programming: multiple goals

# Conflicting usages of water

## Different usages of a multi-purpose dam and control targets:



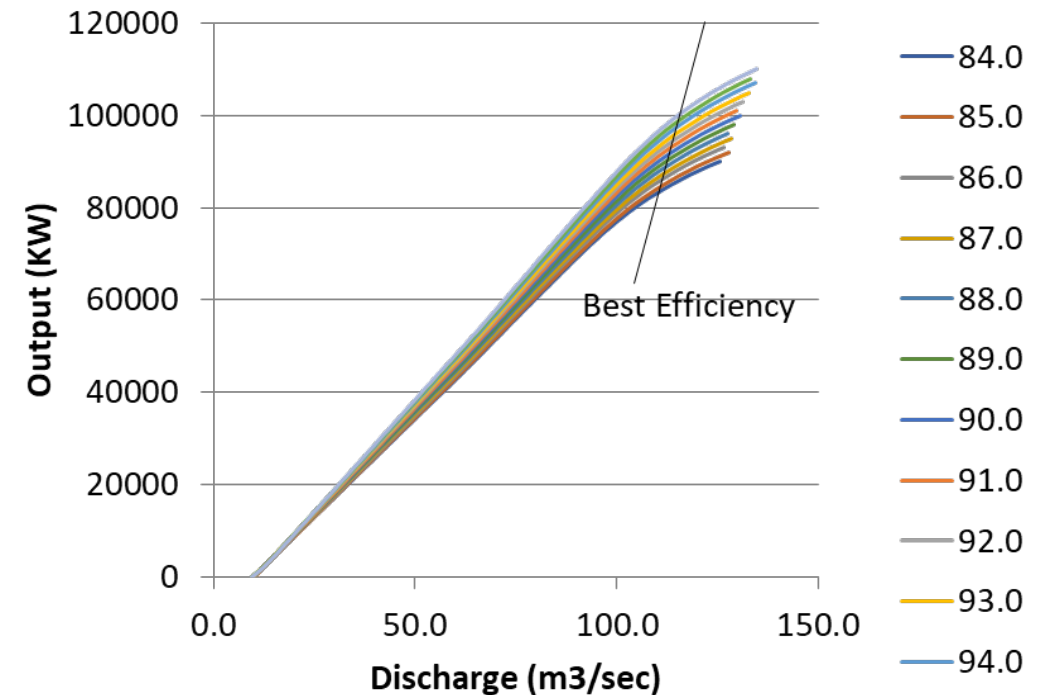
Usage	Operational goal	Priority
Hydro power production	Reservoir level as high as possible Release through turbines	4
Flood protection	Reservoir filling level as low as possible (available flood storage volume as high as possible)	1
Water supply for agricultural, industrial and public supply	Reservoir release, requires sufficient water storage	2
Environmental flow, minimum flow	Reservoir filled sufficiently (as high as possible)	3
Fish protection	Release through spillway	3
Recreation (boating, swimming)	Target water level range	5
Water quality, temperature	Large reservoir volume	2
Limit evaporation losses	Water level as low as possible (surface area as small as possible)	
Groundwater recharge	Continuous outflow	

# Real-world applications

# Niagara Falls (Canada)

Project targets:

- Decision support for operational scheduling
- Unit dispatch
- Storage of water in
  - Grass Island Pool or
  - Pumped Generation Storage
- Simulation models



# Optimizing heat networks

## Heat Networks

Systems that provide heating, cooling or hot water from a central source to multiple consumers

## Design

- What network topology and components and where?
- Will the design be cost-effective in a range of conditions?

## Operation

- How much heat to store and when, considering predicted and uncertain conditions (demand, weather, prices)?

## Input:

MapEditor (ESDL), Modelica or PyCML



Source: WarmingUp

# Hoogheemraadschap van Rijnland

## Model features:

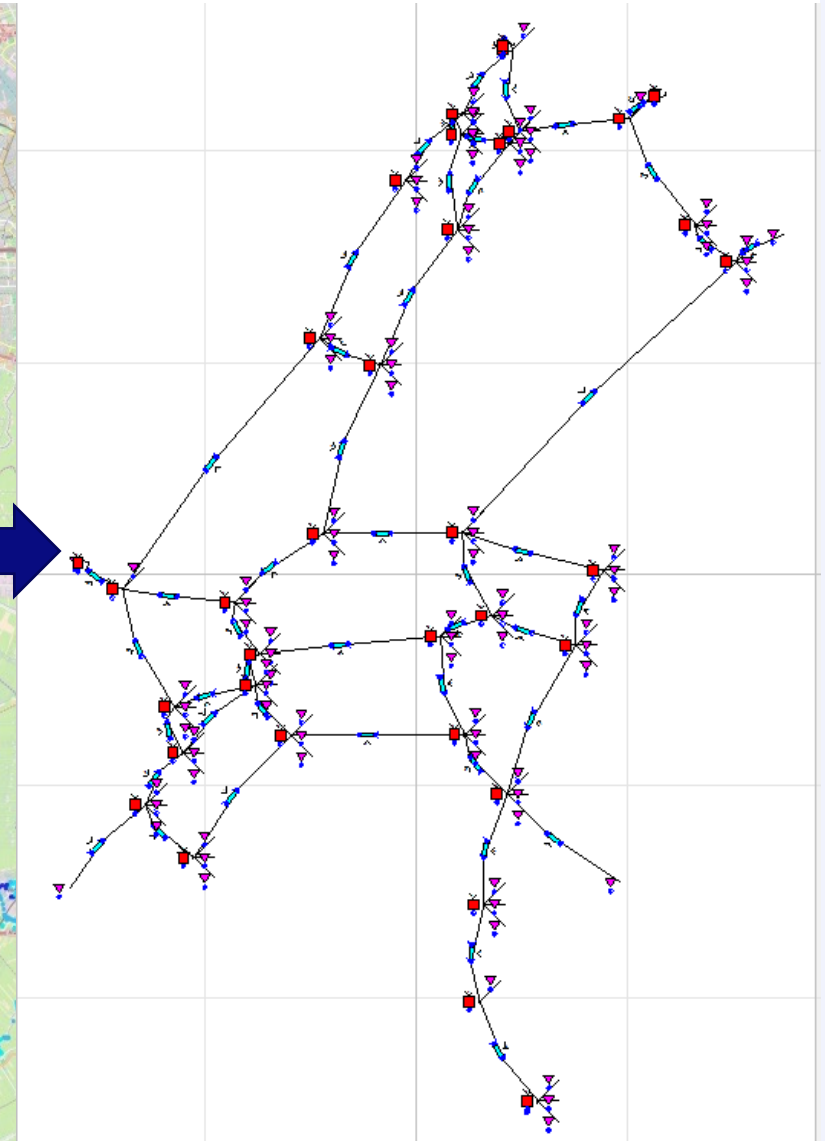
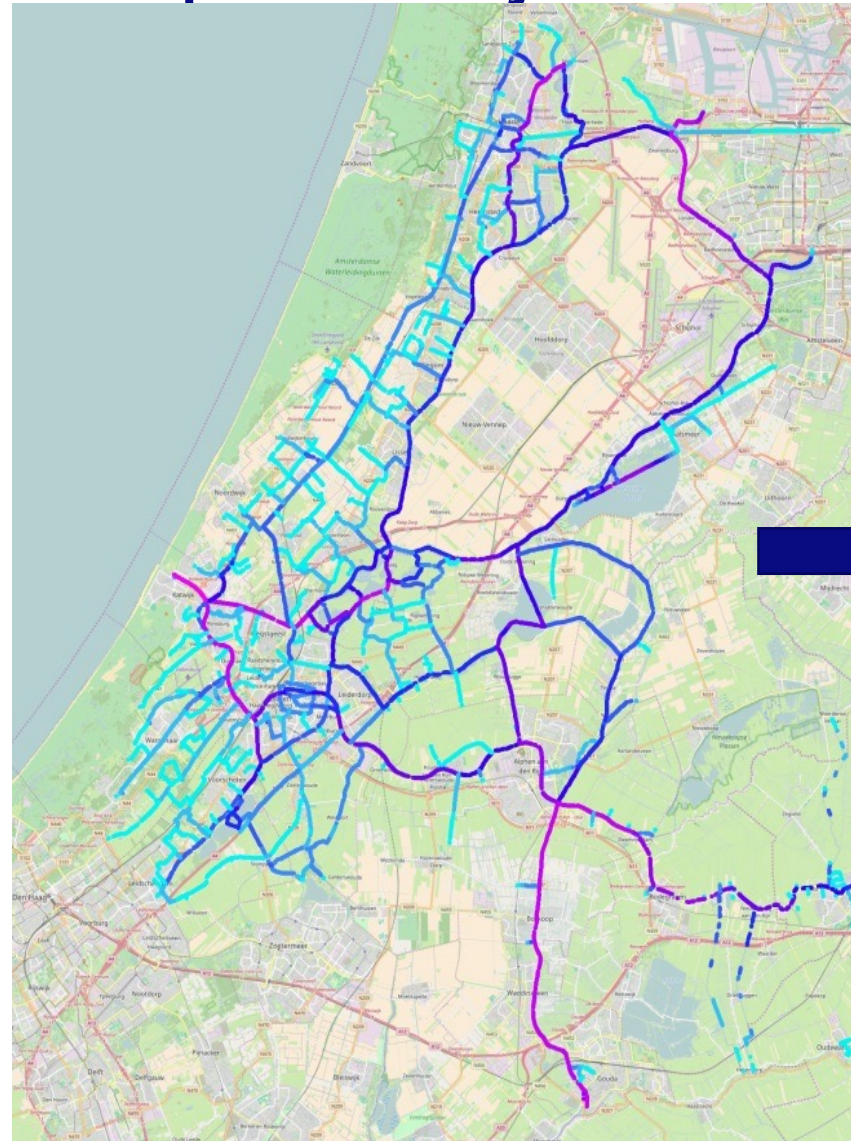
- Non-linear transport equations
- Salt
- Wind
- Pumps
- Wiers
- Large system

## Objective:

***Determine optimal scheduling of hydraulic structures given;***

- Physical operating limits (water levels etc.)
- Operator requests and tidal information
- Best practice operations (minimise switches etc.)
- Energy prices

**Deltares**



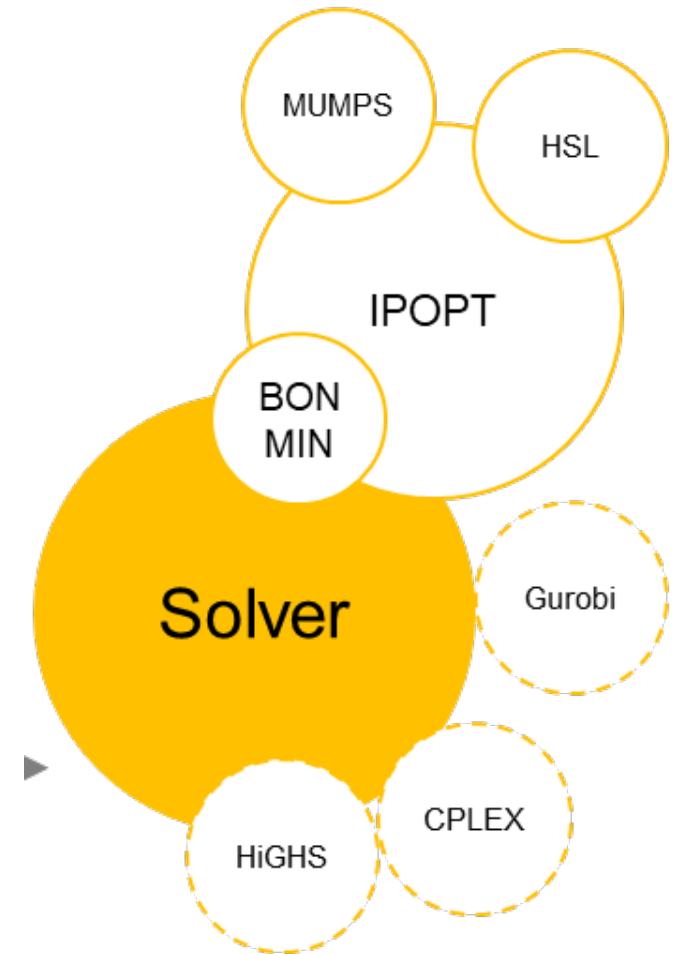
# Mathematic optimization in RTC-Tools

# What does an optimization solver do?

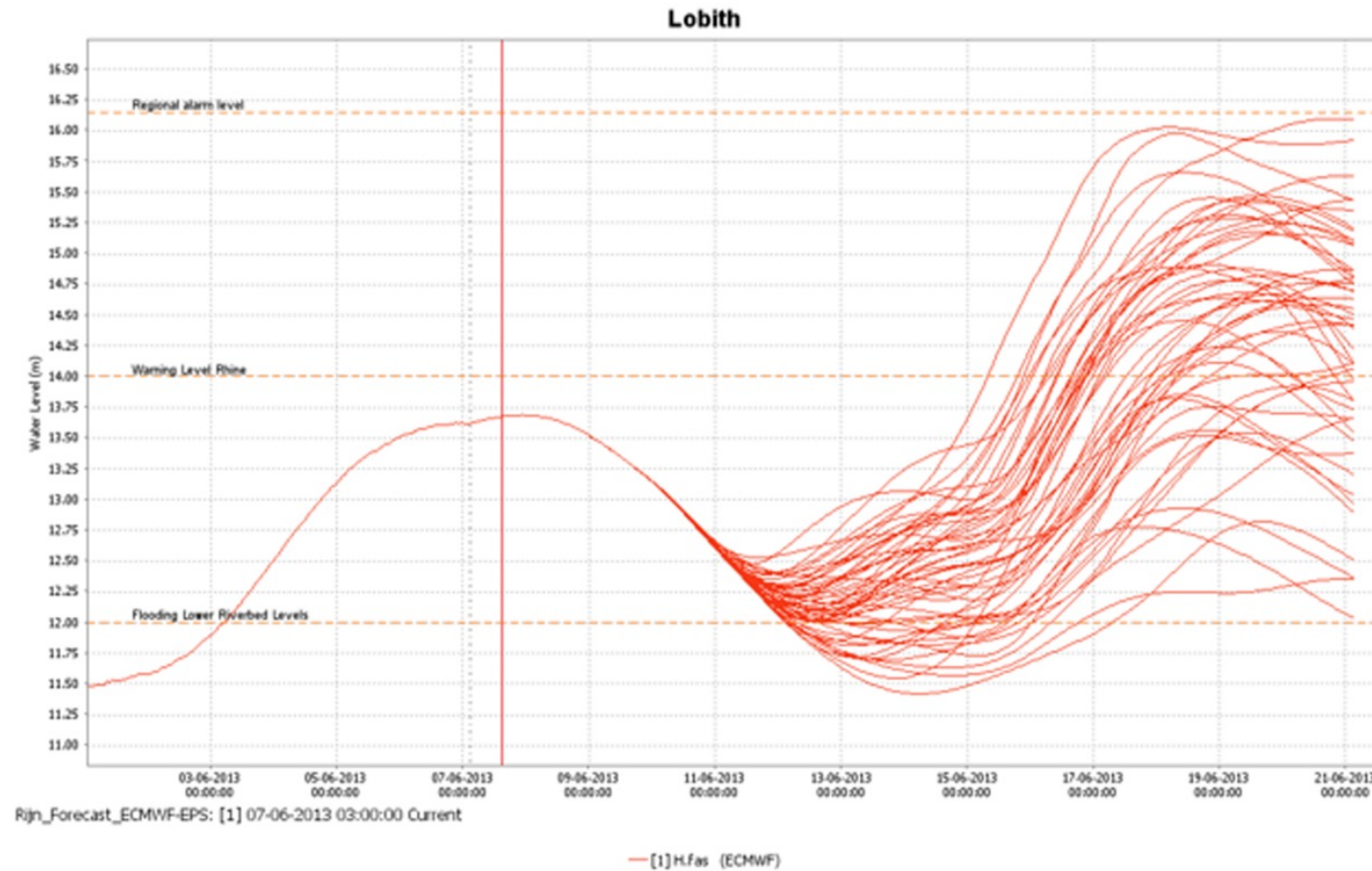
*Implement a mathematical optimization algorithm to solve an optimization problem of a certain form with certain characteristics*

There are many solvers to choose from **and this choice is important**

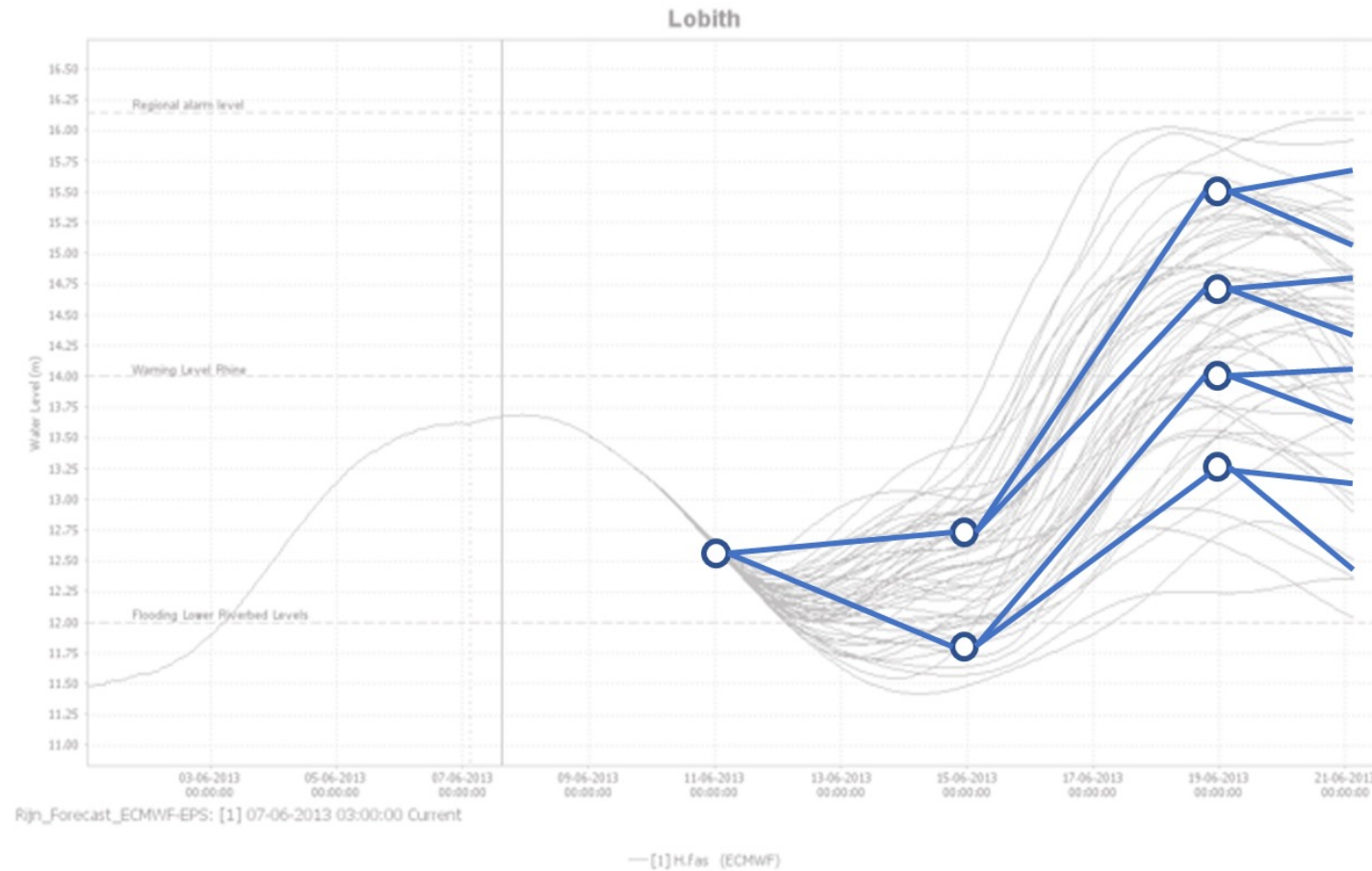
- Is your problem;
  - Continuous?
  - Linear?
  - Mixed Integer?
- Is price important?
- Is speed important?



# Scenario tree example



# Scenario tree example

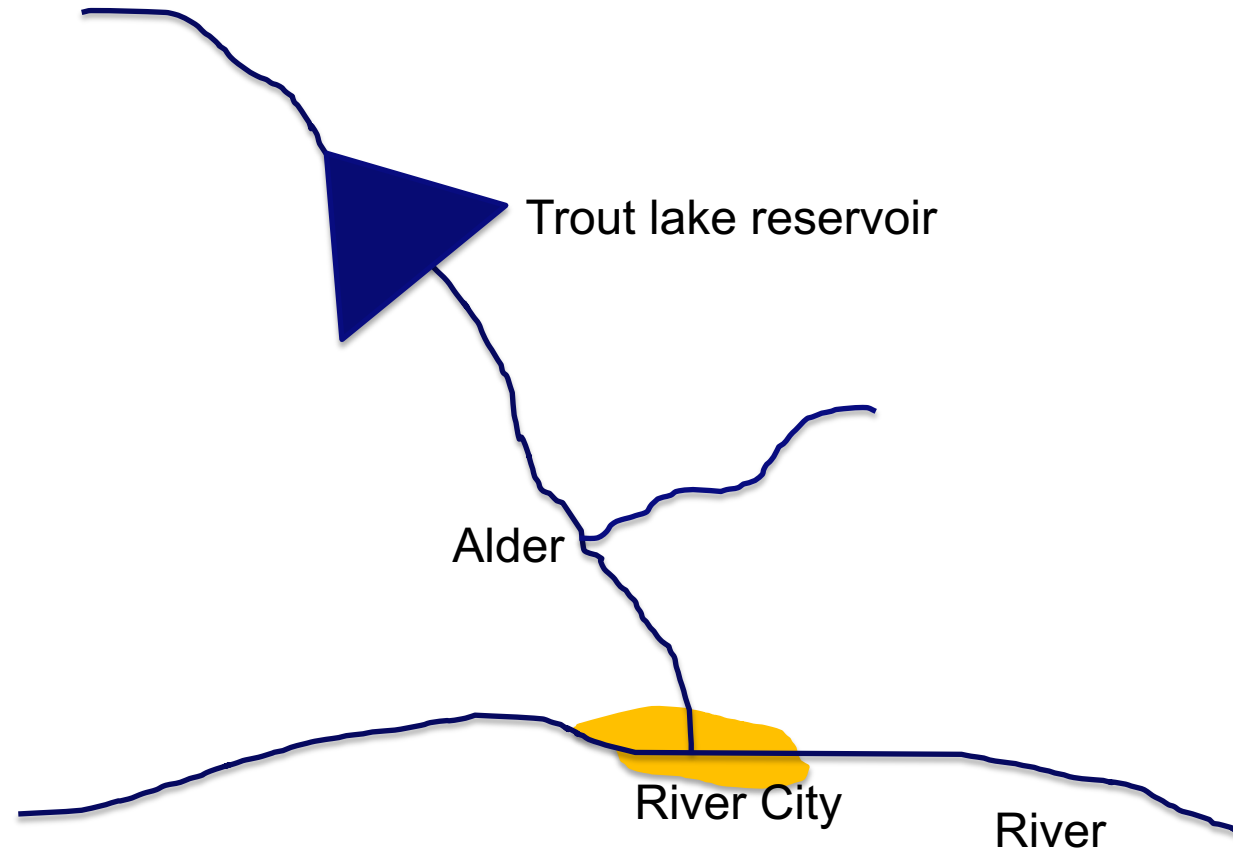


# The Blue River model

RTC-Tools example

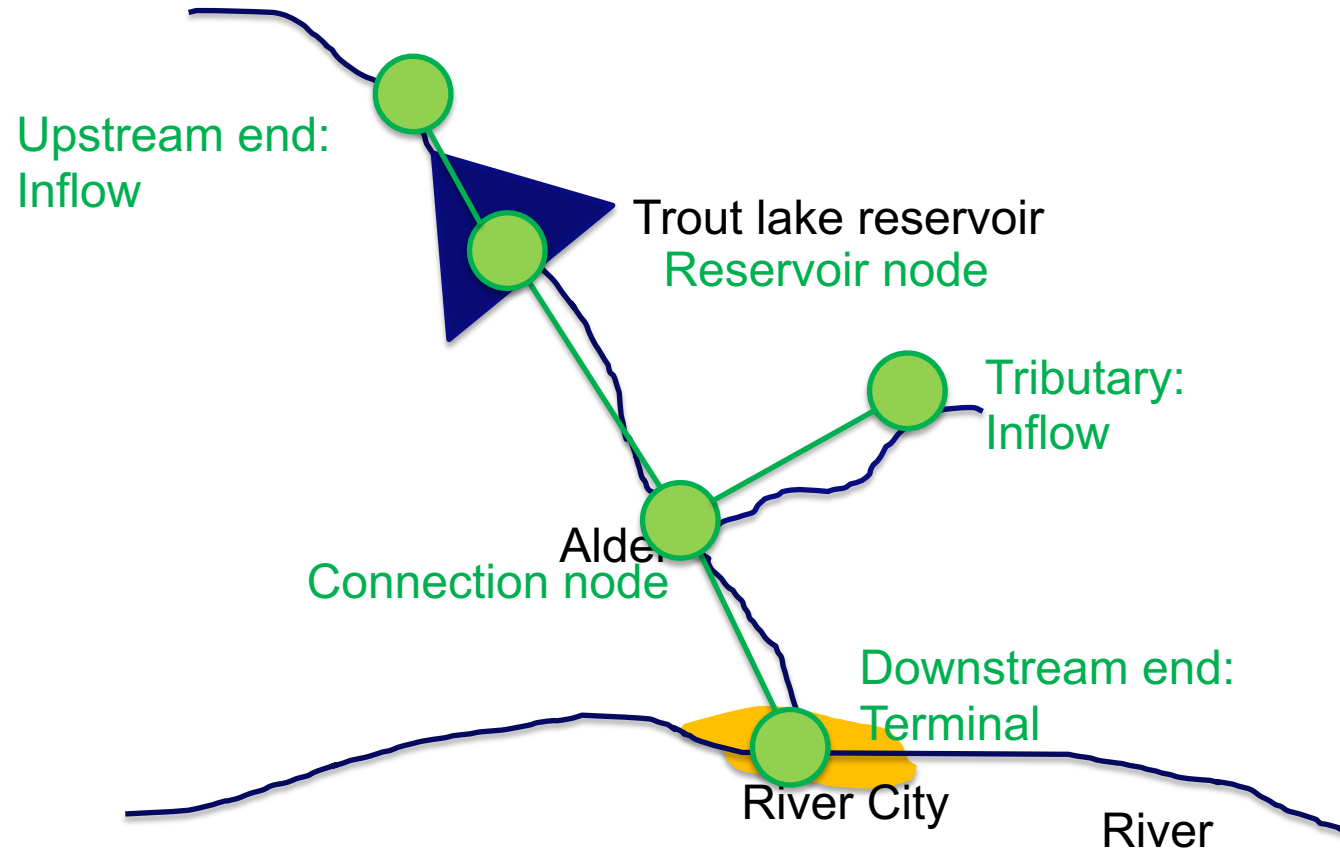
n:\Deltabox\Postbox\Mitchell, Ailbhe\RTCToolsPizzaCourse\

# The study area

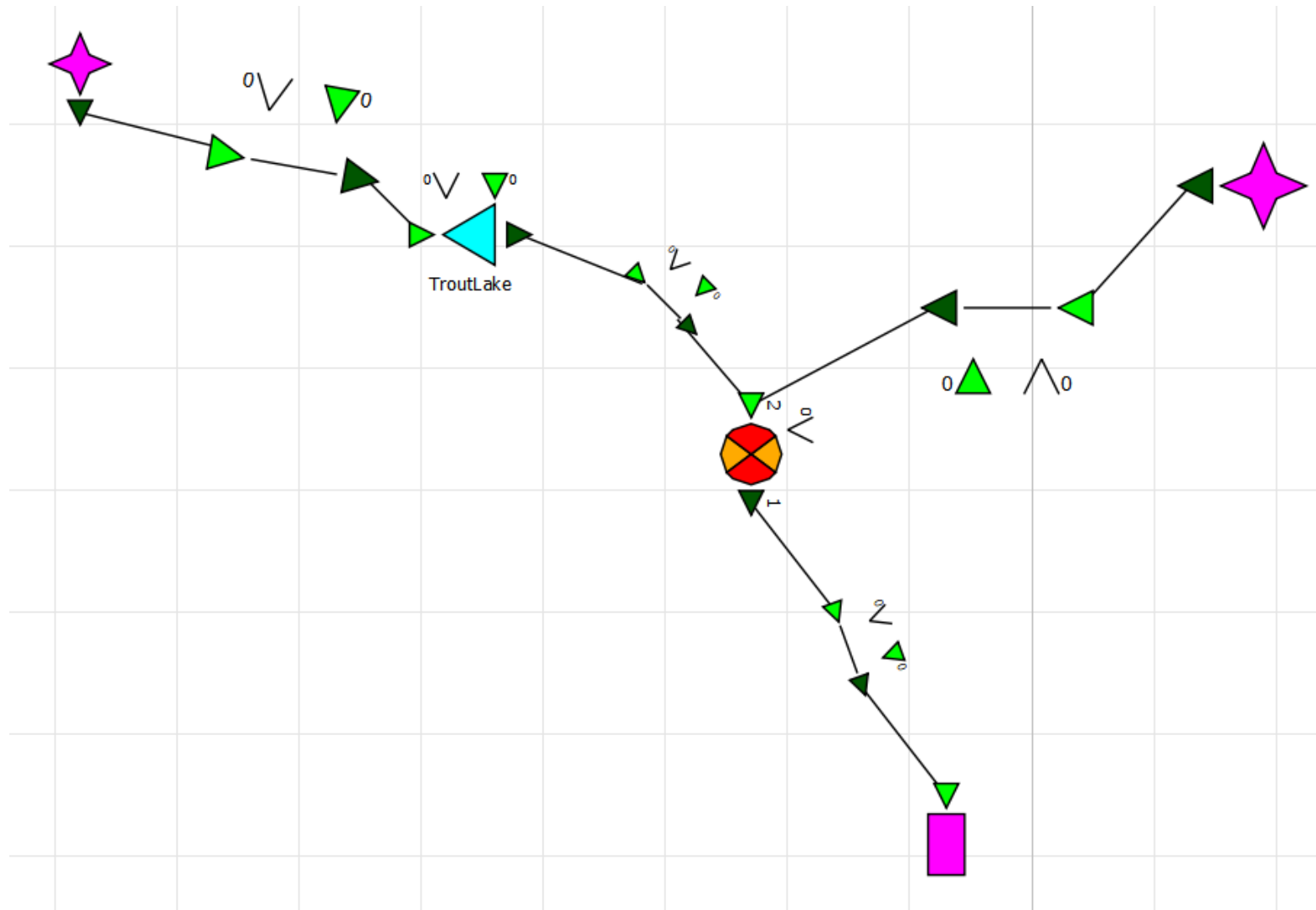


Modified after: O'Connell, S. & Harou, J.: HEC-ResPRM  
Prescriptive Reservoir Quick Start Guide, US Army Corps of Engineers, 2011

# The schematization



# The hydraulic layer: Modelica

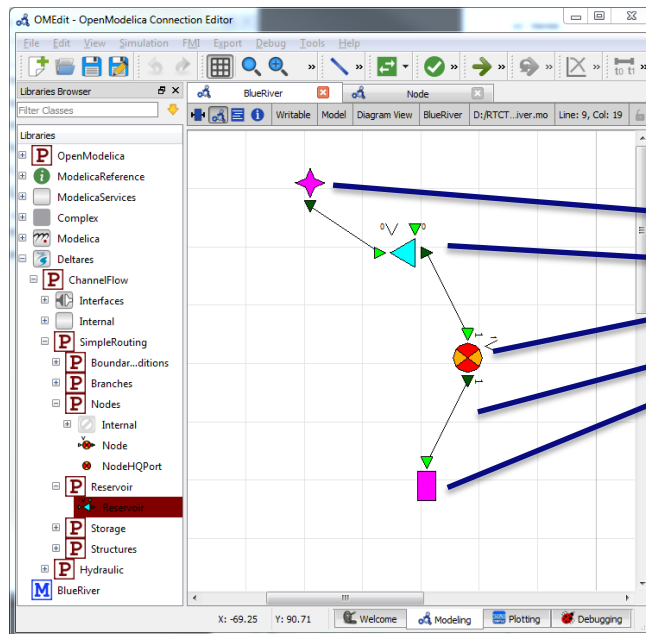


# RTC-Tools

- RTC-Tools components:

Water system model:

- Feeds the optimizer with model parameters **and** derivatives



- Inflow boundary
- Reservoir
- Gauge (River city)
- River section
- Downstream boundary

Control (human interactions on the water system):

- High flexibility for the model setup

Python code for goals

- Release  $Q_{\min}$ ,  $Q_{\max}$
- Reservoir volume min, max

```
# Release recreation rafting
g.append(ForcingRangeGoal(self.get_timeseries('Rafting_Qmin'),
self.get_timeseries('Rafting_Qmax'), 'TroutLake_Q_out', 1, 1.0, [0,100]))
# Storage recreation boating
g.append(ForcingRangeGoal(616740918.8, 629075737.1, 'TroutLake_V', 2,
3.5, [0, 740089102.5285000]))
```

Building blocks for water systems: the Deltares-Modelica library

# Data layer

The screenshot shows the Total Commander 8.52a interface. The left pane displays the directory structure of '020RTC-Tools' under the path '\Courses\2020-04-03NTNU-Optimization\020RTC-Tools\\*. \*'. The right pane shows the contents of the 'output' directory under the path '\2020-04-03NTNU-Optimization\020RTC-Tools\output\\*. \*'. Both panes use a table view to display files and folders.

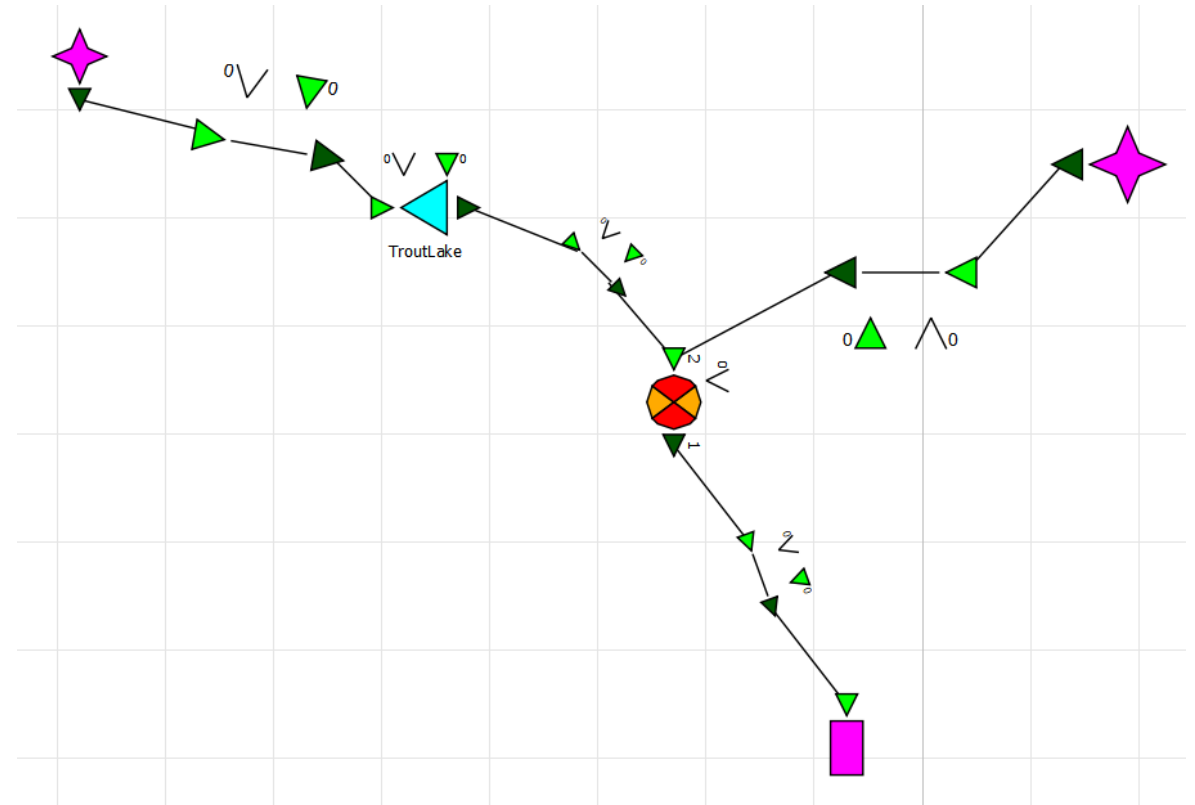
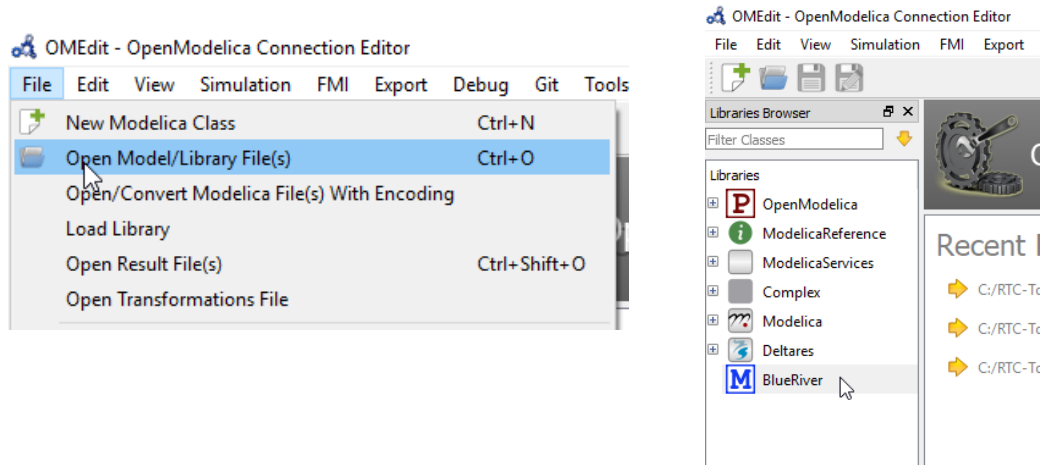
Name	Ext	Size	Date	Attr
..	<DIR>		2020-03-30 21:27	---
_InstallationConfiguration	<DIR>		2020-03-31 22:43	---
input	<DIR>		2020-03-30 21:26	-a-
logs	<DIR>		2020-03-30 21:26	-a-
model	<DIR>		2020-03-30 21:26	-a-
output	<DIR>		2020-03-30 21:26	-a-
src	<DIR>		2020-03-31 22:48	-a-
runBlueRiver	bat	154	2020-03-30 21:27	-a-

Name	Ext	Size	Date	Attr
..	<DIR>		2020-03-30 21:26	-a-
timeseries_export	csv	4 701	2020-03-31 22:48	-a-
timeseries_export	vsz	12 540	2018-11-07 22:01	-a-

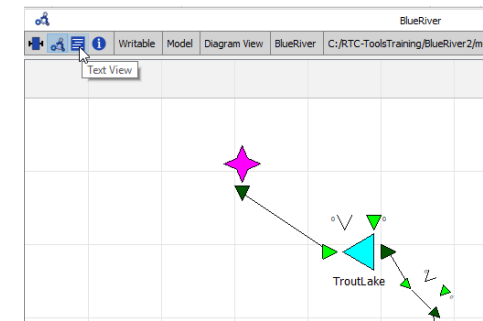
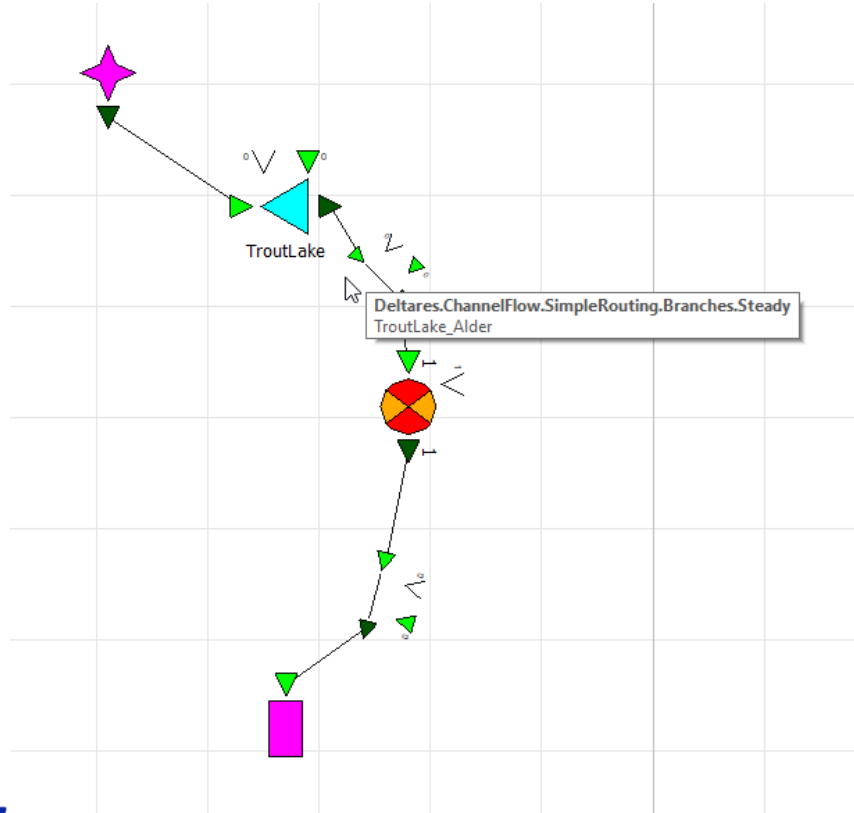
# Open the water system model

1. Open the OpenModelica Connection editor
2. Make sure that the Deltares library is included
3. Open the BlueRiver model file: \model\BlueRiver.mo
4. Double-click on the Blue River Library



# OpenModelica Connection Editor

- Use the mouse-over to identify model objects
- Switch from Diagram View to Text View



```
1 Text View BlueRiver
2 import SI = Modelica.SIunits;
3 // Model objects (nodes)
4 // The inflow boundary condition node for the Trou
5 Deltares.ChannelFlow.SimpleRouting.BoundaryCondi
6 extent = {{-5, -5}, {5, 5}}, rotation = -90));
7 // The reservoir node for Trout Lake. The odd min
8 Deltares.ChannelFlow.SimpleRouting.Reservoir.Reser
9 Q_spill(nominal = 1), Qin.Q(nominal = 0.8), QOut.Q(n
10 extent = {{-5, -5}, {5, 5}}, rotation = 0));
11 // The node that represents Alder
12 Deltares.ChannelFlow.SimpleRouting.Nodes.Node Alde
13 annotation(Placement(visible = true, transformation(
14 // The downstream node, River City
15 Deltares.ChannelFlow.SimpleRouting.BoundaryCondi
16 extent = {{-5, -5}, {5, 5}}, rotation = -90));
```

# Elements of the hydraulic model

## Nodes

- Reservoir
- Bifurcation node
- Terminal
- Inflow
- River branches
- Input
- Inflow time series: external data (as seen by Modelica)
  - Boundary conditions (fixed=true)
  - Optimization results (fixed=false)

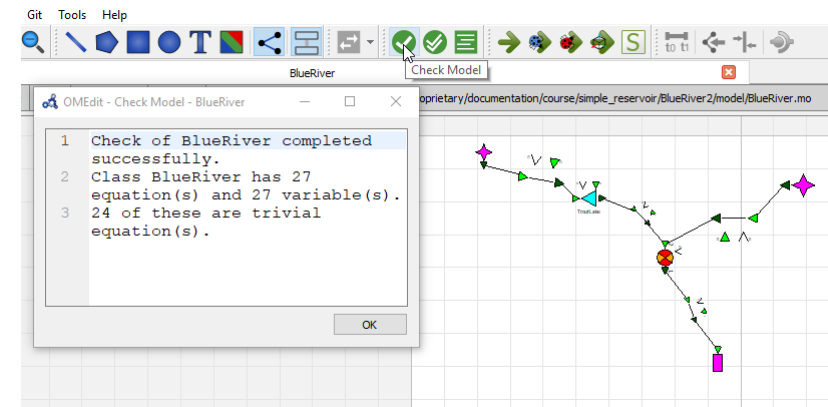
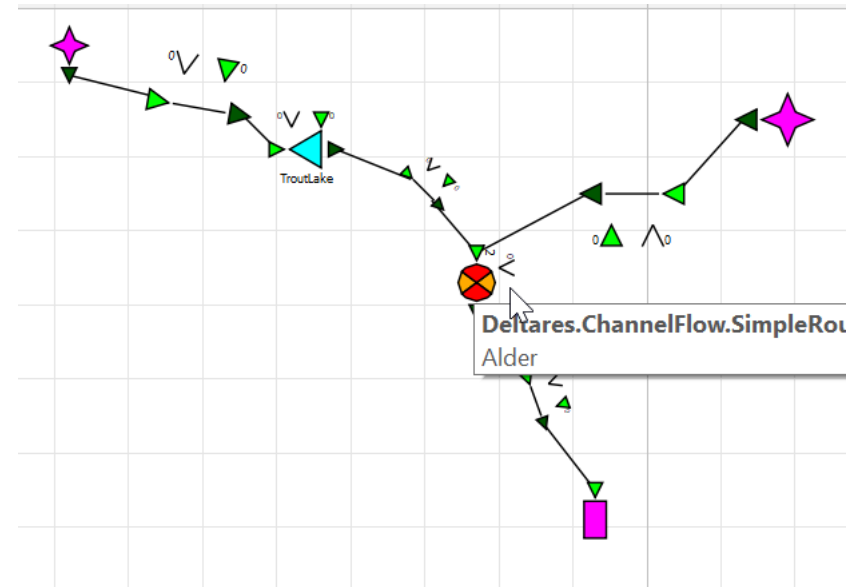
## Output

- Time series computed by Modelica: model results

## Equations

- Connections between nodes
- Assignments of time series
- Other equations

Check the model for a balanced equation system!



# Operational goals for reservoir management



# Modify the python code and analyze the results

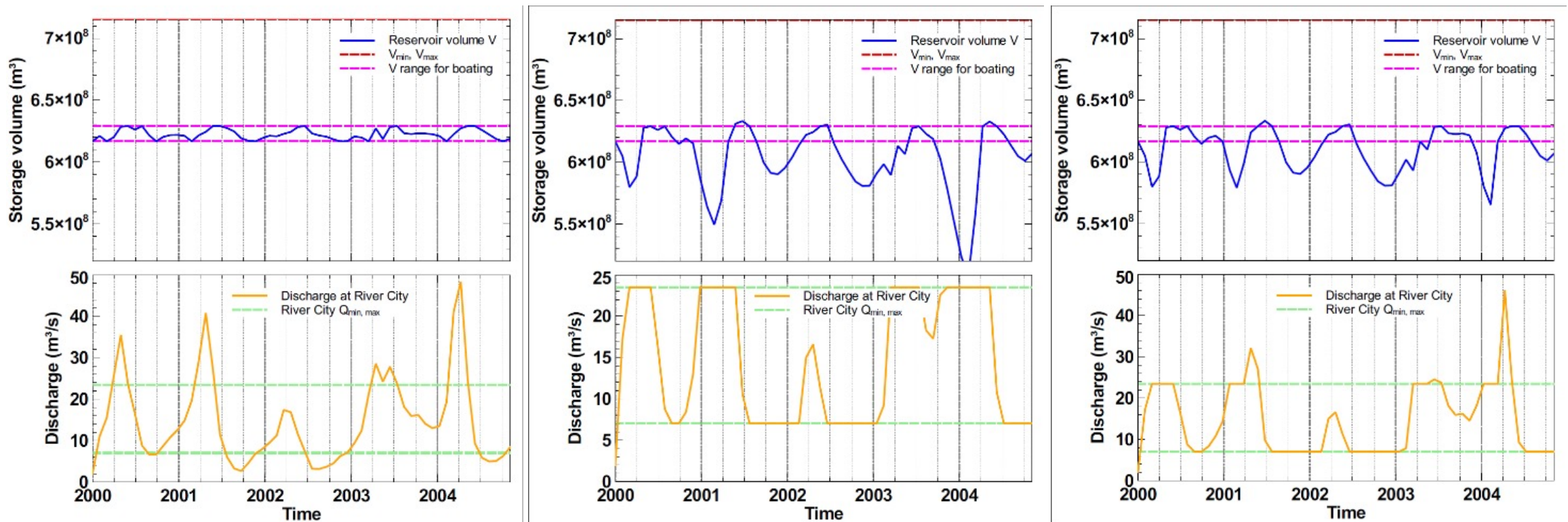
- Run the following three sets and compare the results  
(Only active goals shown)

Goal number		Priority	Weighting factor	Active
020	Recreation boating	2	1.0	True
030	Flow range River City	3	1.0	True

Goal number		Priority	Weighting factor	Active
020	Recreation boating	4	1.0	True
030	Flow range River City	3	1.0	True

Goal number		Priority	Weighting factor	Active
020	Recreation boating	2	0.1	True
030	Flow range River City	2	0.9	True

# Results for different settings



1. Reservoir volume range has priority over discharge, volume range goal is fully met, discharge goal as good as possible under the given conditions (including high priority goal).
2. Discharge has priority over reservoir volume, discharge goal is fully met, volume range not.
3. Compromised solution, both conflicting goals are not fully met, but both are jointly met in a balanced way, here with emphasis on the River City goals.

# References

- [1] Jiada Li, Xiafei Yang, and Robert Sitzenfrei. “Rethinking the framework of smart water system: A review”. In: Water 12.2 (2020), p. 412.

# Contact

 [www.deltares.nl](http://www.deltares.nl)

 [@deltares](https://twitter.com/deltares)

 [linkedin.com/company/deltares](https://www.linkedin.com/company/deltares)

 [info@deltares.nl](mailto:info@deltares.nl)

 [@deltares](https://www.instagram.com/deltares)

 [facebook.com/deltaresNL](https://www.facebook.com/deltaresNL)



**Deltares**