The DSS tools suite: Red-TwoLe

IMRR development relied on an elaborated computer system, named Red-TwoLe, a customized and enhanced version of the RTBR system the Decision Support System (DSS) TwoLe Soncini-Sessa et al. [1999, 2003], developed at Politecnico di Milano, to the RTBR system.

Red-TwoLe system considers two decisional levels: strategic planning and management control. The first aims at defining strategic goals on the long term and includes the design of the policies for the daily operation of the four strategic multi-purpose reservoirs. The second focuses on their management according to the strategic goals defined at the Planning Level.

Red-TwoLe is composed by a combination of different tools, organized into two main modules, which are delivered as ready-to-run virtual machines:

- the Red River Geoportal: a Web-based interface (i) to visualize and explore the results (i.e. hierarchy of indicators, RTBR system models and alternative operating policies), in order to support stakeholders’ participation and negotiation processes at the Planning level; (ii) to support daily release decisions at the Management level;

- the Design Support System: a set of software programs specialized in models identification, policy design and evaluation, to be used at the Planning Level.

Details about these two modules are provided in the next sections.

The integration of Red-TwoLe within the database of Vietnamese Agencies was beyond the IMRR goal and, therefore, the current Red River Geoportal implementation relies on IMRR data only. However, it includes a module to automatically fetch data from external data repositories to update its database in real time. The configuration of this module needs to be customized on the data format of the data providers by the final user of the Geoportal.
0.1 The Red River Geoportal

The Geoportal module of the Red-TwoLe provides a number of services for supporting decision-making activities at both the Planning and at the Management Levels. It indeed offers an effective and non-technical visualization and analysis of the IMRR results across the RTBR system (Fig. 1). The Geoportal is entirely based on Free and Open Source Software (FOSS), and combines enterprise level software such as PostgreSQL (database), Drupal (Content Management System), Apache (web server), Geoserver (map server), Openlayers (map client) with new and fast developing technologies like D3.js (interactive data visualization) and IstSOS (timeseries management for Sensor Observation Services).

0.1.1 Planning Level

At the Planning Level, the Red River Geoportal provides tools to visualize, analyse, and compare the effects of alternative operating policies on the observed/simulated trajectories of different hydrologic variables as well as on the values of the indicators accounting for the interests of the multiple sectors considered. In particular, the Geoportal includes the following functionalities:

- Web map: a geographical map of the RTBR system, which can be navigated by the users to explore data and indicators’ values associated to different locations in the system.

- Project Evaluation Hierarchy: a tree-based chart that illustrates the hierarchy of the 31 evaluation indicators, identified by the main institutional stakeholders as representative of the interests of their respective Sectors.

- Sectors’, Indicators’ and Components’ pages: specific pages providing detailed information that describes (i) the involved Sectors with the list of the indicators they are associated to; (ii) the mathematical formulation of
all the indicators, with interactive tools for exploring their values at different time scales and under alternative operating policies (Fig. 2); (iii) the main components of the RTBR model, including tools for comparatively analysing measured data and models’ output (Fig. ??).

- Scenarios: a switcher for selecting the scenario of interest (e.g., extreme events, socio-economic changes, ...).

- Simulator: an interactive dashboard page (Fig. 4) where, for a given scenario, the user can investigate the system response to a policy (s)he is interested in, by selecting the variables and/or indicators to observe.
0.1.2 Management Level

At the Management Level, the Red River Geoportal allows to obtain in real-time the daily release decisions for the four strategic reservoirs suggested by the Best Compromise Policy on the basis of the current system conditions: the day of the year and the reservoir storages (Fig. 5). When Red-TwoLe will automatically fetch data from the database of Vietnamese Agencies, it will also act as a monitoring system for real-time control of the RTBR systems condition (rain and hydro gauging stations, reservoirs levels, hydropower production, etc., see Fig. 6) and the current values of all the indicators.

0.2 The Design Support System

This module is constituted by a set of four software suites which allow designing and updating the operating policies for the RTBR system.

The Red River Simulator Suite is a collection of three simulators operating at different levels of accuracy:

\[1\text{.i.e. the efficient policy that the Final Political Authority will chose}\]
• Distributed simulator: it is the most accurate simulator. It provides a spatially distributed representation of the Red River delta, based on a 1D hydrodynamic model (MIKE 11). However, the complexity of this model makes it very slow\(^2\) thus allowing the simulation of only few user selected alternative.

• Evaluation-oriented simulator: it is less accurate than the previous simulator. It is based on a set of matlab/octave functions and is able to estimate the 31 evaluation indicators under a user-selected scenario for all the efficient alternatives (thousands).

• Design-oriented simulator: it is the least accurate but the fastest simulator. It is implemented in C++, in order to reduce its computational requirement, as the policy design process requires to run millions of simulations. It is based on \textit{yaml}, a human friendly data serialization standard for programming languages.

The Optimization Suite is composed by two optimization engines: the MOEA Framework, which was employed for the design of the operating policies for Son La, Hoa Binh, and Tuyen Quang reservoirs via Evolutionary Multi-Objective Direct Policy Search, and the TwoLe Planning Engine, used for the optimal operations of Thac Ba.

The MOEA Framework is a free and open source Java library for Multi-Objective Evolutionary Optimization that can be combined with the Design-oriented simulator for designing Pareto optimal operating policies. The architectural choice of separating MOEA Framework and the simulator represents a significant improvement with respect to the common monolithic optimization tools, where the system model and the optimization routine are packed together. In fact, the simulator can be easily and independently updated (for example by

\(^2\)It takes 2 days for simulating 16 years
adding a new system components such as a new reservoir) without requiring to modify the optimization engine.

The TwoLe Planning Engine provides an alternative optimization engine relying on Stochastic Dynamic Programming, which can be used for designing optimal operating policies only in small/medium systems. For this reason, this engine cannot scale to the entire RTBR system, while it was suitable for the design of the optimal operating policy for Thac Ba reservoir.

The Interactive Parallel Plot Suite is an interactive visualization tool that allows exploring the performance of multiple alternative solutions on a multidimensional plot, thus supporting the negotiation process with the institutional stakeholders for the identification of good compromise solutions.

Finally, the IMRR General Toolbox is an Octave-based set of tools that addresses specific project needs:

- Time series storage, management and conversion to Red River Geoportal format;
- Analysis of Mike11 simulations;
- Identification of reduced models by means of Artificial Neural Networks (ANN);
- Within daily and within turbines optimization of hydropower plants;
- Generation of synthetic and extreme inflow scenarios;
- Downscaling of climate change scenarios;
- Input data pre-processing for Interactive Parallel Plot;
- Estimate of the flood-condition via a Fuzzy Empirical Classifier;
- Simulation of Reservoir Management Guidelines (both 2011 and 2015 versions).
Chapter 1

The Red River Geoportal

The following pages contain the user manual of the Red River Geoportal.
Red River Geoportal Manual

Author: Carolina Arias, Marco Micotti, Enrico Weber
Supervisor: Rodolfo Soncini Sessa
Date: 11-11-2015
## Contents

1. INTRODUCTION .................................................................................................................. 7  
   1.1. Glossary ......................................................................................................................... 7  
   1.2. Geoportal architecture ................................................................................................. 9  
   1.3. Geoportal users ........................................................................................................... 11

2. GEOPORTAL MANUAL ........................................................................................................ 12  
   2.1. Markers Pop-ups ........................................................................................................... 12  
   2.2. Browse and edit Markers ............................................................................................ 15  
   2.3. Browse and edit Sectors ............................................................................................ 19  
   2.4. Browse and edit Indicators ......................................................................................... 21  
   2.5. Visualizing the Project Hierarchy ............................................................................... 22  
   2.6. Special pages ............................................................................................................... 23  
      2.6.1. Dashboard pages .................................................................................................... 23  
      2.6.2. Compare charts ..................................................................................................... 25  
      2.6.3. Project Hierarchy management ............................................................................. 26  
      2.6.4. Simulation page (for advance analyst, admin) ....................................................... 27  
   2.7. Add & edit contents (analyst, advance analyst, admin) ............................................ 29  
      2.7.1. Add Sector / Add Marker / Add Indicator ............................................................ 29  
      2.7.2. Add level ................................................................................................................ 29  
      2.7.3. Update marker from a shape file ......................................................................... 30  
      2.7.4. Create a Scenario ................................................................................................. 31  
      2.7.5. Import observations from .csv file ...................................................................... 33  
      2.7.6. Create dashboard page .......................................................................................... 33  
      2.7.7. Add user manual page .......................................................................................... 33  
   2.8. Dealing with geographical data (admin) .................................................................... 35  
      2.8.1. Add layers into the IMRR geoportal .................................................................... 35  
      2.8.2. Add layers to the main map ............................................................................... 36  
      2.8.3. Style your layers ................................................................................................. 37  
   2.9. User management (admin) ......................................................................................... 40  
      2.9.1. Add a new user ...................................................................................................... 40
2.9.2. Manage permissions ........................................................................................................ 40

3. OTHER RESOURCES AND USEFUL LINKS ..................................................................... 44

Annex A. Matlab structures specification ................................................................................. 45

Annex B. SLD styles ................................................................................................................ 47
## Table of figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Geoportal architecture</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>IMRR Geoportal front page</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>Component Marker popup: description</td>
<td>13</td>
</tr>
<tr>
<td>4</td>
<td>Component Marker popup: variable list</td>
<td>13</td>
</tr>
<tr>
<td>5</td>
<td>Sector Marker popup</td>
<td>14</td>
</tr>
<tr>
<td>6</td>
<td>Indicator content popup</td>
<td>14</td>
</tr>
<tr>
<td>7</td>
<td>Marker list page</td>
<td>15</td>
</tr>
<tr>
<td>8</td>
<td>Marker edit page</td>
<td>15</td>
</tr>
<tr>
<td>9</td>
<td>Map edit controls</td>
<td>16</td>
</tr>
<tr>
<td>10</td>
<td>Add Marker pop up</td>
<td>16</td>
</tr>
<tr>
<td>11</td>
<td>Marker’s Observations view example</td>
<td>17</td>
</tr>
<tr>
<td>12</td>
<td>Marker’s procedures view example</td>
<td>18</td>
</tr>
<tr>
<td>13</td>
<td>Sector list page</td>
<td>19</td>
</tr>
<tr>
<td>14</td>
<td>Sector page example</td>
<td>19</td>
</tr>
<tr>
<td>15</td>
<td>Add sector/level pop up</td>
<td>20</td>
</tr>
<tr>
<td>16</td>
<td>Add sector/level pop up 2</td>
<td>20</td>
</tr>
<tr>
<td>17</td>
<td>Indicators list page</td>
<td>21</td>
</tr>
<tr>
<td>18</td>
<td>Insert Observations from csv</td>
<td>21</td>
</tr>
<tr>
<td>19</td>
<td>Project hierarchy</td>
<td>22</td>
</tr>
<tr>
<td>20</td>
<td>Dashboard Pages list</td>
<td>23</td>
</tr>
<tr>
<td>21</td>
<td>Dashboard Page example</td>
<td>24</td>
</tr>
<tr>
<td>22</td>
<td>Editing Dashboard Page</td>
<td>24</td>
</tr>
<tr>
<td>23</td>
<td>Compare charts page</td>
<td>25</td>
</tr>
<tr>
<td>24</td>
<td>Synchronous two-panel chart</td>
<td>25</td>
</tr>
<tr>
<td>25</td>
<td>Project hierarchy list links</td>
<td>26</td>
</tr>
<tr>
<td>26</td>
<td>Simulation Page</td>
<td>28</td>
</tr>
<tr>
<td>27</td>
<td>Add content pop up</td>
<td>29</td>
</tr>
<tr>
<td>28</td>
<td>Update marker from shape file</td>
<td>30</td>
</tr>
</tbody>
</table>
Figure 29. Add Service tab ........................................................................................................ 31
Figure 30. Select service tab ...................................................................................................... 32
Figure 31. Create manual page pop up .................................................................................... 34
Figure 32. Upload image ........................................................................................................... 34
Figure 33. Import shape file ..................................................................................................... 35
Figure 34. Openlayers /layers tab ............................................................................................ 35
Figure 35. Openlayers map tab ................................................................................................ 36
Figure 36. Edit map section ....................................................................................................... 37
Figure 37. Edit map section/overlay layers .............................................................................. 37
Figure 38. Add a new style ......................................................................................................... 38
Figure 39. People ........................................................................................................................ 40
Figure 40. Permissions ............................................................................................................... 41
Figure 41. User “demo” roles .................................................................................................... 42
Figure 42. Node view permission section ................................................................................. 43
Welcome to IMRR Geoportal Documentation

The “Red River Geoportal”¹ is a web tool, GIS-based, created to access all the information produced by the IMRR project. The project adopts the Multi Criteria Decision Analysis (MCDA now on) methodology to generate, analyze and compare different planning alternatives for the management of the Red River water system in Vietnam. The Geoportal includes georeferenced information related to Red River System models, sectors and indicators, as well as a number of analysis tools (dynamic charts, two panel charts, and custom dashboard) and a simplified user interface to run custom simulations.

The “Red River Geoportal” has been developed within the IMRR Project by Hydro Informatics Lab and Natural Resource Management group at Politecnico di Milano.

IMRR (Integrated and sustainable water Management of Red-Thai Binh River System in a changing climate) Project has been co-funded by Italian Ministry of Foreign Affairs, Politecnico di Milano and Institute of Water Resource Planning (IWRP) of Vietnam.

¹ In this document often referred as “Geoportal”
1. INTRODUCTION

1.1. Glosary

Hereafter a list of terms used in the present document is specified, with a brief definition:

Criterion: aspect considered by at least one stakeholder as relevant to assess alternative performance.

Sector: group of stakeholders that share the same interests on the system and the same point of view.

Indicator: mathematical formulation to assess the effects, expressed in physical units, of the alternatives for a given criterion.

Project Hierarchy (Evaluation hierarchy): tree diagram reporting hierarchically:

- Sector at the first level. In the IMRR Project: Energy deficit, Floods control, Water supply, Environment, Navigation.
- Sector’s Criteria: in the intermediate Levels of the Hierarchy
- Indicators are the last level (leaves of the tree diagram)

Model component: part of the global model of the system describing the mode of operation of a specific component, use to compute time trajectory of remarkable variables for each component.

Time series: generic time series dataset, it can be related either to variable or indicator. It corresponds to the Procedure, in the IstSOS jargon.

Variables: generic physical measurement, i.e. water level, streamflow, volume.

Alternative: outcome of simulation for a given scenario and management policy. In a gauge station a variable can be the historical value measured, or the simulated value for a given alternative. It corresponds to the Observed Properties, in the IstSOS jargon.

Scenario: coherent dataset of variables and indicators, can be selected through the Sidebar and its visibility can be customized for each user. It corresponds to the Service, in the IstSOS jargon and corresponds to a specific schema in the underlying database.

Marker: point of interest represented on the map, it corresponds to the FOI (Feature of interest), in the IstSOS jargon. There are two types of marker:

- Component marker: represent a model component, i.e. hydropower plant, reservoir, gauge station.
- Sector marker: represents a specific sector and give access to indicator list for the specific marker.
  There can be more than one marker for each sector, depending on spatial distribution of problems represented by the sector itself.

Shapefile: geographic information file format, defined by ESRI and commonly used to exchange geographic data. It is composed by different files, usually zipped together.

Feature: item of a content type

Geometry field: location and geometry information related to a give feature (i.e. lat/lon for a marker)
1.2. Geoportal architecture

The Geoportal has been developed like a “light weighted” SDI, which components are reported in Figure 1 and describe hereafter. In the backend, the Database Management Systems PostgreSQL with the spatial extension PostGIS is used to store all data, including time series and geospatial data. Geoserver is used as the GIS Server, creating and managing several types of OGC Web Services such as Web Map Service and Web Feature Service. IstSOS1 server allows the managing and dispatching observations from monitoring sensors and other points of interest according to the OGC Sensor Observation Service standard. IstSOS connects also with the PostgreSQL database using a distinct data schema for each scenario considered in the project.

The client side is composed by Drupal CMS (Content Management System), taking advantage of the existing Geoserver and OpenLayers modules (similar to Cartaro distribution) and developing two new modules:

- a general use module, to connect istSOS and Drupal and to manage multiple time series and different scenarios inside Drupal;
- a project-customized module, to manage interaction between map, pop-up, time-series charts and search functionalities.

These modules, allow to interact with the different software within the Drupal application, without need to access directly to software such as OpenLayers or istSOS to manage data, because all the main functionalities are available thought the user-friendly interface.

Being a CMS, Drupal offers the possibility to have different user accounts and permissions (i.e. general public, project stakeholders and project owners), gaining additional advantage such as the extendibility,
namely the possibility to create different plugins, to organize and query contents and information within the website.
1.3. Geoportal users

The documentation is geared toward 2 distinct types of users: the common user or stakeholder, and the Geoportal advanced user or analyst. A super user level has been also defined, with administrative rights on all the software on the top of which Red River Geoportal has been built, but it will not be described in this manual.

**Stakeholders** have registered access to:

- Map and GIS layers
- Indicator and Sector Popup and related content
- Dashboard and compare charts pages

**Analyst** have registered access to:

- Map and GIS layers
- Indicator and Sector Popup and related content
- Dashboard and Compare charts pages
- Add content functionalities
- Project hierarchy management
- Custom Simulation run

2. GEOPORTAL MANUAL

The following diagram describes some of the functionalities available in the front page of the IMRR Geoportal:

![IMRR Geoportal front page](image)

1) **Map area.** View the map, the markers and other geographical features.
2) **The layer switcher.** Use this to set the base map and to display markers and geographical features on the map.

3) Side tool bar (or Side bar). Use this to:
   a. **search** markers, **filtering by sector or component,**
   b. **browse** Geoportal contents, visualizing Project Hierarchy, Markers, Sectors, Indicators and Procedures lists and to
   c. **select** scenario to be shown in the charts.

4) **User menu.** Use this to access the Add content functionalities, to Special pages, to Manuals pages and to your account details. You can also exit the user mode by clicking on Log out.

5) **Admin Toolbar.** Use this to have a quick access to manage all functionalities of Drupal CMS. The present manual will not deal these topics, please refer to the Drupal documentation: [https://www.drupal.org/documentation/modules/toolbar, https://www.drupal.org/project/toolbar](https://www.drupal.org/documentation/modules/toolbar, https://www.drupal.org/project/toolbar)

2.1. **Markers Pop-ups**

Clicking one of the markers on the map, a new window will pop up with information related to the specific marker. Information can vary depending on the type of marker (sector of component).

For the component marker, the pop up is divided in two parts:

- on the top, the component description (see Figure 3)
- at the bottom, a dynamic time series chart, that allows to choose among a list of all the variables associated to the component and to select and compare all the time series computed for that variable with the selected scenario (see Figure 4)

---

**Figure 3. Component Marker popup: description**
For the sector marker, the pop up is divided in two parts (see Figure 5).

- on the left side, there are the list of indicator related to the specific marker and the list of all the other indicators related to another marker (and location) of the same sector;
- on the right side, the project hierarchy sub-branch, related to the specific sector

Clicking one of the item of indicators list, the content related to the specific indicator will be shown in the pop-up (see Figure 6):

- on the top part, indicator formulation
- at the bottom a dynamic time series chart, where it is possible to choose between different time frequency of indicator computation and to select among all alternatives computed for that indicator and scenario.
Figure 6. Indicator content popup
2.2. Browse and edit Markers

On the sidebar is possible to access a list of all markers (Marker list page, see Figure 7). Clicking on the marker name, you will enter the Marker page.

![Figure 7. Marker list page](image)

On the Marker edit page you will find five tabs:

![Figure 8. Marker edit page](image)

1) **View.** View marker’s location on the map.

2) **Edit.** On this tab you can set or change the marker’s *Title, label, type* (Sector, Component) and its *short label*; add a short or long *Description*.

   **NOTE:** *Title* must be unique in the Geoportal, the *label* field will be shown on the map, *short_label* field is the key value to join variables and indicator to markers.

![Figure 9. Map edit controls](image)

In the Marker type select box, choose the marker type: ‘sector’ or ‘component’. If you choose the Marker to be a Sector it will automatically appear in the Sector marker list field.
The marker pop up provide an OpenLayers editor in order to create geographical entry on the map. You can edit marker geometry on the map using the controls on the right upper corner of the map:

- The **Draw point** button allows to create a marker point on the map. If the marker already exists, the tool will overwrite the new given position.

- The **Modify geometry** button allows you modify shape of a feature. For polygon markers, with this tool you can change the shape of the polygon. For point markers, this tool simply moves a feature.

- In case the marker is a polygon, you can create a hole on it using the **Draw a hole** button.

- The **Drag geometry** button allows you to simply move a feature.

- The Select geometry button allows you to select a feature, delete it, split it or merge more than one feature.
The Snapping settings button, you can change the snapping tolerance value. The snapping tolerance is the distance within which the pointer or a feature will snap to another location, using *Draw point* or *Modify geometry* tools. If the location being snapped to (vertex, boundary, midpoint, or connection) is within that distance, the pointer will automatically snap. Snapping tolerance is measured using pixels units.

Navigation button is the same as the pan button.

3) *IstSOS observation View*. To visualize the markers’ related time series (variables for component marker, indicators for sector marker), divided by scenarios, you can go to the IstSOS observations view (see Figure 11). In order to have a quick view to time series content, it is possible to select it (a) and then the type of chart (b).

![Figure 11. Marker’s Observations view example](image)

4) *IstSOS procedures list*. Here you can see the details (general info, name, sensor ID, description, location, etc...) of all procedures related to the marker. It is possible to delete time series.
Figure 12. Marker's procedures view example
2.3. Browse and edit Sectors

On the sidebar is possible to access a list of all sectors, and the links take you to the sector description (see Figure 13) and a list of markers inside a specific sector.

![Figure 13. Sector list page](image)

**Floods Control**

**Description**

Floods sector takes into account damages related to floods in Red River basin. Areas affected by river floods involve 15 provinces, including Ha Noi, with a population of 18.8 million. The most critical area, where highest damages are expected if a flood should occur, is Hanoi city, which has therefore been considered as the reference section for IMR project indicators.

In order to complete information about damages all over the interested area, there are indicators related also to floods effect at Pha Lai, Tuyen Quang and along the Day River, in case of opening of the Van Coc/Day diversion system.

**Markers inside the Floods Control sector:**

<table>
<thead>
<tr>
<th>Short label</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>FL_Day</td>
<td>view</td>
</tr>
<tr>
<td>FL_HN</td>
<td>view</td>
</tr>
<tr>
<td>FL_TQ</td>
<td>view</td>
</tr>
</tbody>
</table>

![Figure 14. Sector page example](image)

Clicking on the link `view` on the markers list, will take you directly to the related marker page.

Clicking on the Edit tab is possible to change Sector description and title, to associate Sector Markers and Sectors, to customize position and label of the Sector in the Project Hierarchy.
Figure 15. Add sector/level pop up

On the Marker List you can choose which marker associate to this new sector/level.

Figure 16. Add sector/level pop up 2

After this in the Menu settings section check the Provide a menu link box, type a title for the menu link (equal to the sector Title) and a description. Leave the Parent item as <Project Hierarchy> (Sector is the upper level of the tree).
2.4. Browse and edit Indicators

On the sidebar is possible to access a list of all indicators (see Figure 17) and a brief description. Clicking on the indicator name, you will enter the indicator page where you can modify the indicator description and fields, similarly to Sector page.

Note that Indicators can be automatically created uploading a time series from .csv files and clicking on the Indicator checkbox (see Figure 18).

![Figure 17. Indicators list page](image1)

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Short description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOD &gt; 10</td>
<td>The BOD value at Lien Mac, Nhue river larger than 10 mg/l.</td>
<td>BOD value at Lien Mac &gt; 10 mg/l</td>
</tr>
<tr>
<td>city of Hanoi</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DAY Opening</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Regulation No 62/1999/ND-CP of the Prime Minister states that the Van Coc/Day dam system can be used to divert floods when water level in Hanoi reaches 13.4m and the forecast of Center for Hydro-Meteorological Forecast shows an increasing flooding level. Moreover, Decree No. 04/2011/ND-CP of the Government stipulates that the maximum flow drainage of Day river is limited to 2,500 m³/s, and that the Day dam will be used only to prevent floods with returning time of at least 500 years in Red river system, or in case of serious incident on dyke system in Hanoi.

The operational rule of the Day dam, according to the above regulations can be summarized in the IARR model (Note that the modelling time step is 1 day, so the real rule is here simplified to match with this time-step), as follows:

1. If \( \Delta a_{t+1}^{DN} < 3.4 \) then the Van Coc/Day dam system is closed and \( Q_{DN}^{UP} = \frac{Q_{DN}^{UP}}{DN} \) m³/s.
2. If \( \Delta a_{t+1}^{DN} \geq 3.4 \) and \( a_{t+1}^{DN} > 12.8 \) then the system is open and \( Q_{DN}^{DN} = Q_{DN}^{UP} - Q_{DN}(a_{t+1}^{DN}) \),

being:

- \( a_{t+1}^{DN} \) the tomorrow forecasted level in Hanoi,
- \( Q_{DN}^{UP} \) and \( Q_{DN}^{DN} \) are the average daily flow in Red River, respectively, upstream and downstream of Van Coc section on day \( t+1 \)
- \( Q_{DN}(a_{t+1}^{DN}) \) is a function reproducing the state-discharge relationship between the flow diverted by the Van Coc/Day system and the forecasted water level in Hanoi, represented in figure.

![Figure 18. Insert Observations from csv](image2)
2.5. Visualizing the Project Hierarchy

On the sidebar is possible to access a popup with the project hierarchy tree chart, where it is possible to interactively explore sectors and criteria. Clicking on sectors (red boxes) give direct access to Sector page inside the same pop-up.

![Project Hierarchy Diagram]

Figure 19. Project hierarchy
2.6. Special pages

To access to the special pages contents click the link Special pages on the users menu.

2.6.1. Dashboard pages

Dashboards allows to compare and analyze many variables at the same time, considering all the alternatives on different time horizons. Clicking on the dashboard pages entry of the Special Page menu gives the list of dashboards created using (see Create dashboard page). Clicking on the page title you are interested in, it will open the dashboard page as it is shown in Figure 20.: a custom number of panels reporting variables and indicators, on different time periods, with the possibility to add or remove interactively alternatives from the dropdown menu.

<table>
<thead>
<tr>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation Output</td>
</tr>
<tr>
<td>test graph page</td>
</tr>
<tr>
<td>Hoa Binh Reservoir dashboard</td>
</tr>
<tr>
<td>Simulation</td>
</tr>
<tr>
<td>Water level Hanoi</td>
</tr>
</tbody>
</table>

Figure 20. Dashboard Pages list
Clicking on edit tab gives the possibility to customize the dashboard choosing which time series (both variable and indicator) add, the chart style (bar or lines), and the time period.
2.6.2. Compare charts

On the Compare Charts page is possible to compare variables or indicators using two panel charts with synchronized time slider. You have to drag and drop variables or indicators into the two boxes “Graph 1” and “Graph 2”, and then their values will be plotted in the chart below.

Figure 23. Compare charts page

Figure 24. Synchronous two-panel chart
2.6.3. Project Hierarchy management

Clicking on Project Hierarchy management entry, all the elements of the Hierarchy are presented in a table as a structured list (see Fig. ...): here it is possible to quickly reorder levels, dragging and dropping each element.

**NOTE:** each element of the hierarchy is stored in Drupal as a link related to a node (Sector, Level or Indicator).

Edit and Delete functionalities, as well as the Add Link tool of Project Hierarchy management page, change properties of the link related to the node but not the node itself.

To add Sector, Level or Indicator node refer to Add Sector/ Add Marker / Add Indicator.

![Figure 25. Project hierarchy list links](image-url)
2.6.4. Simulation page (for advance analyst, admin)

This page allows users to simulate the whole Red River system with user-defined input, namely:

- a time interval, specifying dates when to start and stop the simulation (maximum length: 365 days)
- a custom inflow time series (Son La, Hoa Binh lateral, Yen Bai, Thac Ba, Ham Yen, Tuyen Quang), that has to be uploaded in the inflow file form below, following this template (link)
- the initial level for each one of the 4 reservoirs (Son La, Hoa Binh, Thac Ba, Tuyen Quang), has to be set in the form below

Before running the first simulation, each user has to create also a new Chart Page, using “Simulation Output” as page name, selecting the variables and indicators to be shown after the simulation run.

![Select start date](image)

![Select stop date](image)

![Initial reservoir levels](image)

![Inflow input file](image)

Figure 26. Simulation Page
Clicking on the “Run” button, IMRR Geoportal simulates reservoirs releases according to the management policies corresponding to the interesting alternatives identified in the last IMRR Basin Meeting (namely XX, YY, ZZ), given the provided inflows and for the defined time period, computing also the effects both in term of water levels and streamflows in the system, and in term of indicators.

Once Simulation ends, results are directly uploaded into Geoportal and the “Simulation Output” chart page is loaded in the Geoportal. Simulation results can be therefore accessed selecting through the combo menu the alternatives with "_sim" suffix (i.e. XX_sim).
2.7. Add & edit contents (analyst, advance analyst, admin)

To add and edit contents click the menu link Add content on the users menu. A popup window will appear showing the contents that can be added:

![Add content pop up](image)

2.7.1. Add Sector/ Add Marker / Add Indicator

These entry are shortcuts to the edit tab of Marker, Indicator and Sector page (see Browse and edit Markers. Browse and edit Markers, Browse and edit Sectors.Browse and edit Sectors; Browse and edit Indicators.Browse and edit Indicators).

2.7.2. Add level

1) Click on Add Content / Add Level,

2) Fill up field Title, representing level name

3) Flag “Provide a menu link” and fill up the menu link title field: it is the label of the level (or criteria) in the Project Hierarchy.

4) Choose a suitable Parent Item, between the existing elements of Project Hierarchy

5) Save

The level content is displayed inside the Project Hierarchy with its Title - to make it be shown inside the Project Hierarchy at the right place, check the Provide a menu link box and select the place inside the Parent item select box.
2.7.3. Update marker from a shape file

If you have a point shape file of your marker, you can simply create a Marker by uploading a zipped shape file. Go to Add content / Update marker from shape file.

Click on choose file, choose the zipped shape file and then click on import. Once the zipped file is uploaded, a marker is created with the same title as the name of the shape file and finally added to the map. Then you can edit it going to the Marker list link and using the edit link (see Browse and edit Markers).

NOTE: marker shapefile must have same fields of marker node.
2.7.4. Create a Scenario

A scenario corresponds to a dataset, also called Service in the istSOS jargon: clicking on Add content / Create Scenario, Geoportal link to the istSOS configuration module. To create a new scenario (see Figure 29. Add Service tab)

1) Enter a Scenario (Service) Name.

   NOTE: on the underlying database a service corresponds to a schema: only single word with lowercase letters are admissible as name;

2) Scenario name is used also in the csv files that contains time series to join a time series to a given scenario.

3) Enter the Scenario (Service) Description: it is the description that will be shown on the sidebar, in the scenario selector area (see Figure 2. IMRR Geoportal front page)

4) Enter EPSG code: the Reference System of the Feature of Interest layer, namely marker layer in the “Red River Geoportal”, which has EPSG:32648.

5) Click on Select service to activate the new scenario, selecting the Feature of Interest layer

6) To enable users to see scenarios different from the default scenario, please refer to Manage permissions.

---

**Figure 29. Add Service tab**

**Figure 30. Select service tab**
2.7.5. Import observations from .csv file
This functionality let to upload or update time series in the geoportal (see Browse and edit Indicators):

1) choose and upload a file with time series data

2) if you are uploading an indicator:
   a. make sure that the field name reported in the .csv file before the '_' (underscore) matches with the title of some existing indicator node in the Drupal;
   b. check “override existing description” if you want to update indicator short description field or, in case the indicator does not exist yet as Drupal node, to create it.

3) Click verify and insert data: Geoportal will check the csv header before upload data, create the new time series (procedure) and then add data.

**NOTE:** uploading an existing time series (either variable or indicator) will update data related to existing time period and it will add data related to nonexistent time periods.

Number of alternatives (columns in the csv) must be always the same for a given variable. The only way to add new alternatives is to delete existing time series (see ...) and upload again all the data.

The csv file structure is detailed into annex A, as well as a procedure to massive import csv data from a given folder.

2.7.6. Create dashboard page
Shortcut to edit tab of dashboard page (see Dashboard pages).

2.7.7. Add user manual page
In case you want to add a user manual page, go to Add content / Add user manual page. You can insert a Title (Manuals) and write the text on the body.
In case you need to add an image to the text, go to IMAGE to upload an image file (only png, gif, jpg and jpeg are allowed)

When you finish click on Save.
2.8. Dealing with geographical data (admin)

2.8.1. Add layers into the IMRR geoportal

To add other geographical data, different from the markers, into the map (i.e. cities - points -, basin - polygon -) is possible to use the Openlayers and Geoserver modules in Drupal. The first thing you need to do is to import a shape file in Geoserver. Go to Add content / Import shape file in Geoserver.

![Add content]

**Import shape file in Geoserver**

Look for your zipped shape file by clicking on Choose file, then click on Upload and then Import. Then you will be redirected to the Layers page of the OpenLayers Module. The layer imported in this way is created by default as a Web Map Service WMS layer (see http://www.opengeospatial.org/standards/wms). If you want to served your layer as WFS, after you import the layer, you can go to Structure / OpenLayers / Layers, click on Edit and change layer type from WMS to WFS.

![OpenLayers /layers tab]

Here you can click on Edit on the layer you just added, and edit it (change the name, projection, type...)

2.8.2. Add layers to the main map

To add any layer of the layers list (Structure / OpenLayers / Layers) into the main map, you need to go to the Admin Toolbar: Structure / OpenLayers / Maps
Find the **analyst_map** and click on **Edit**.

Click on **Layers and styles**. Go to Overlay Layers.
Search for the layer you want to add to the map and click on “Enabled” and “In switcher” if you want to add it to the layer switcher on the map. If you want your layer to be visible when you open the map, click on “Activated”. Go to the end of the page and click on Save.

2.8.3. Style your layers

For WMS layers

If the layer you want to style is a WMS layer you can use the Styled Layer Descriptor Language SLD 1.0. version (http://www.opengeospatial.org/standards/sld). See Styling in Geoserver documentation (http://docs.geoserver.org/2.4.x/en/user/styling/index.html - remember to check the documentation of the Geoserver version you are using).

Navigate to Structure / GeoServer / Styles and click Add to create a new style for use with GeoServer. Choose a name of your liking and provide a description about the style. Enter the SLD source code in the text field provided and click Save.

You can write the SLD by hand but it is usually more favorable to make use of tool support. QGIS (http://www.qgis.org/) is the most common tool used but other style editors should be fine as well.
One you create the style, now you can go to the layer you added: Structure / OpenLayers / Layers, click on Edit, Styles and write the name of the style you created. Then you should see the layer on the map styled accordingly with the sdl file you created.

For WFS layers

If the layer you want to style is a WFS layer you can proceed as for the WMS layer case. First you need to consider the following: There is a problem of different SLD support between Drupal, GeoServer and OpenLayers. GeoServer and OpenLayers are only supporting SLD 1.0 version, while Drupal supports SLD 1.1 version. This means that SLD files as produced by QGIS can't simply be used in Drupal without adjustments. However due to the similarity of SLD 1.0 and SLD 1.1 the amendments are usually a matter of stripping namespaces and renaming tags starting with Svg so that they read Css. It's best to evaluate yourself which route to go and which process works best for you. Check Annex B. SLD styles to see the SLD styles already implemented on the IMRR geoportal.
2.9. User management (admin)

Clicking on People link in the administrative bar give access to all the Drupal tools for user management. To understand permissions and roles in Drupal, please refer to the official documentation, here https://www.drupal.org/node/120614:

2.9.1. Add a new user

Navigate into the Admin Toolbar: People, and then click on Add user. You will be prompt to the People pop up where you can create a username and password of your new user, as well as establish the most appropriate role. When you finish click on Create new account.

![People](image)

**Figure 39. People**

2.9.2. Manage permissions

To specify the permissions for a role, click The Admin Toolbar: People / Permissions. There you can give to the users’ roles the ability to edit any page within the site. Scroll down the permissions page and click the checkbox next to 'edit any page content'.
NOTE for user management in “Red River Geoportal”:

1. In order to customize scenario visibility for each user, Geoportal automatically creates a role for each scenario. Default scenario is always visible for all users, whereas to grant visibility to a specific scenario, the role with the scenario name has to be given to that user. In Figure 41. User “demo” roles, for example, the user “demo”, can see scenarios “demotest” and “imrrhistorical”.

<table>
<thead>
<tr>
<th>PERMISSION</th>
<th>ANONYMOUS USER</th>
<th>AUTHENTICATED USER</th>
<th>ADMINISTRATOR</th>
<th>ANALYST</th>
<th>DEMOTEST</th>
<th>IMRRHISTORICAL</th>
<th>SIMULATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Climate stations Create new content</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Climate stations Edit own content</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Climate stations Edit any content</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Climate stations Delete own content</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Climate stations Delete any content</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Discharge station Create new content</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Discharge station Edit own content</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Discharge station Edit any content</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2. Stakeholder users cannot see dashboard pages created by other user: this setting is tuned flagging the component type in Node View Permission page under Configuration/Content Authoring page and then customizing access on the People/Permission page, Node View Permission section (see Figure 42)
Select content types configurable by extra permissions "View own content" and "View any content".

- Area
- Article
- Catchment
- Cities
- Climate stations
- Discharge station
- Graph page
- Indicator
- International bound
- Irrigation district
- Level
- Main roads
- Manuals
- Markers
- Marker polygon
- Basic page
- Project branch
- Provincial boundary
- Pump
- RR basin fao
- Sector

Figure 42. Node view permission section
3. OTHER RESOURCES AND USEFUL LINKS

Screencasts:
- add sector marker and edit its properties
- create a new sector
- create a new indicator for the new sector
- upload data for the indicator
- use compare charts
- create and edit dashboard page
- run simulation and check results

Presentations:
- [http://www.slideshare.net/NRMPolimi/foss4-g2015-ariasmicotti](http://www.slideshare.net/NRMPolimi/foss4-g2015-ariasmicotti)

Other links:
- [https://www.drupal.org/](https://www.drupal.org/)
- [http://geoserver.org/](http://geoserver.org/)
- [http://istsos.org/](http://istsos.org/)
- [https://www.gnu.org/software/octave/](https://www.gnu.org/software/octave/)
### 4. Annex A. Matlab structures specification

Data uploaded in the Geoportal must follow specification below.

1. An header part, composed by 13 “key, value” lines:

<table>
<thead>
<tr>
<th>Field name</th>
<th>Field type</th>
<th>Description</th>
<th>Geoportal use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>String</td>
<td></td>
<td>It must be unique in the Geoportal because it is used for the procedure name on the istSOS side. It must match the name of the file. <strong>Variables:</strong> typeofvariable_location_model(ifany) ex. q_ST_model: streamflow simulated at Son Tay <strong>Indicator:</strong> nameofindicator_frequency(ifany) ex. dmgHN_d: Indicator of Hanoi damages, daily values <strong>NOTE:</strong> nameofindicator should match with an existing indicator node on Drupal. If not, Geoportal can create it for indicator, but it works only for indicators related to all the horizon (= no “_frequency” in the name)</td>
</tr>
<tr>
<td>Author</td>
<td>String</td>
<td>Reference person for the specific file</td>
<td></td>
</tr>
<tr>
<td>Created</td>
<td>Date/String</td>
<td>Date of creation</td>
<td></td>
</tr>
</tbody>
</table>
| id_type    | String     | Type identifier | **Variables:** TS_type of variable  
*TS_v:* volume.  
*TS_h:* height (water level)  
*TS_q:* streamflow  
*TS_en:* energy production  
*TS_t:* temperature  
*TS_x:* rainfall  
*TS_ev:* evaporation  
**Indicators:** always equal to 1 |
| marker_id  | String     | Marker identifier | Used to join time series and marker node. It is composed by 2 elements, divided by an underscore “_”:  
*componenttype_place.* It must be unique, in the geoportal, in order to obtain the related istSOS FOI (Feature of interest)  
ex. res_HB: Hoa Binh reservoir component marker  
*FL_HN:* Flood Hanoi sector marker |
<p>| description| String     | time series description | Short description of the time series, it is used to load the description in istSOS, as label for the list of the Sector pop-up and as title of the Indicator page and Pop- |</p>
<table>
<thead>
<tr>
<th>Field name</th>
<th>Field type</th>
<th>Description</th>
<th>Geoportal use</th>
</tr>
</thead>
<tbody>
<tr>
<td>unit</td>
<td>String</td>
<td>Unit of Measurement</td>
<td>Used as label in the y-axis in the time series dynamic chart.</td>
</tr>
<tr>
<td>time_sampling</td>
<td>String</td>
<td>Type of time series: interval or instantaneous</td>
<td></td>
</tr>
<tr>
<td>origin</td>
<td>String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>scenario</td>
<td>String</td>
<td>Scenario name</td>
<td>Must match with an existing scenario (service) name. Must follow PostgreSQL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>schema naming constrains ex. “IMRRhistorical”, “climatechange”</td>
</tr>
<tr>
<td>frequency</td>
<td>String</td>
<td>Possible values:</td>
<td>Variables: always daily</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Indicator: horizon (corresponds to all period in the scenario), yearly,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>monthly, daily</td>
</tr>
<tr>
<td>calendar</td>
<td>String</td>
<td>Always equal to datenum, it indicates that dates</td>
<td>.csv are in matlab/octave format.</td>
</tr>
<tr>
<td>interpolator</td>
<td>String</td>
<td>Optional field, used only for parameters</td>
<td></td>
</tr>
</tbody>
</table>

1. The data matrix [N rows, M columns], where
   
   - matrix first row (line 14 on .csv file) contains M labels, with comma separator
   
   - column 1, contains dates, in matlab format
   
   - column from 2 to end and rows from 2 to end, contain data
5. Annex B. SLD styles

The styles used on the IMRR geoportal can be resumed in 6 types:

1. Polygon Fill, Border and transparency
2. Polygon Fill, Border, transparency and labels
3. line
4. Complex line
5. line with labels
6. Points with graphic

For reference, go to: [http://docs.geoserver.org/2.4.x/en/user/styling/index.html](http://docs.geoserver.org/2.4.x/en/user/styling/index.html). On the following table you can find the present layers on the analyst map with the style type and sld files:

<table>
<thead>
<tr>
<th>Layers name</th>
<th>Layer switcher name</th>
<th>Style type</th>
<th>Sld file name</th>
</tr>
</thead>
<tbody>
<tr>
<td>A. IMRR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sluicegates</td>
<td>IMRR: Sluicegates</td>
<td>Points with graphic</td>
<td>A1_sluicesgates</td>
</tr>
<tr>
<td>canal_pump</td>
<td>IMRR: Canal Pumps</td>
<td>Points with graphic</td>
<td>A6_pump</td>
</tr>
<tr>
<td>Rivers_polygon</td>
<td>IMRR: Rivers</td>
<td>Polygon Fill,Border and transparency</td>
<td>A1_rivers</td>
</tr>
<tr>
<td>irrigation_districts</td>
<td>IMRR: Irrigation Districts</td>
<td>Polygon Fill,Border, transparency</td>
<td>A1_irrrdistrits</td>
</tr>
<tr>
<td>tide_gs</td>
<td>IMRR: Tide Gauging Station</td>
<td>Points with graphic</td>
<td>A6_tidegs</td>
</tr>
<tr>
<td>network</td>
<td>IMRR: Network</td>
<td>line</td>
<td>A3_network</td>
</tr>
<tr>
<td>B. Sand mining survey</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sandmining_site</td>
<td>Sand mining survey:</td>
<td>Polygon Fill,Border and transparency</td>
<td>B1_sandmining site</td>
</tr>
<tr>
<td>sandmining_barge</td>
<td>Sandmining survey:</td>
<td>Points with graphic</td>
<td>B6_Sandmining_Barge</td>
</tr>
<tr>
<td>C. Hydrology</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>climate_st</td>
<td>Hydrology: Climate stations</td>
<td>Points with graphic</td>
<td>C6_climate_station</td>
</tr>
<tr>
<td>catchment</td>
<td>Hydrology: Catchment</td>
<td>Polygon Fill,Border and transparency</td>
<td>C1_cachment</td>
</tr>
<tr>
<td>waterlevel_station</td>
<td>Hydrology: Water level stations</td>
<td>Points with graphic</td>
<td>C6_waterlevel_station</td>
</tr>
<tr>
<td>rainfall_station</td>
<td>Hydrology: Rainfall stations</td>
<td>Points with graphic</td>
<td>C6_rainfall_station</td>
</tr>
<tr>
<td>Layers name</td>
<td>Layer switcher name</td>
<td>Style type</td>
<td>Sld file name</td>
</tr>
<tr>
<td>-------------------------</td>
<td>------------------------------------------</td>
<td>---------------------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>discharge_station</td>
<td>Hydrology: Discharge stations</td>
<td>Points with graphic</td>
<td>C6_discharge_station</td>
</tr>
<tr>
<td>D. Basemap</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cities</td>
<td>Basemap: Cities</td>
<td>Points with graphic</td>
<td>D1_cities</td>
</tr>
<tr>
<td>provincial_boundary</td>
<td>Basemap: Provincial Boundaries</td>
<td>line</td>
<td>D2_provincial_boundary</td>
</tr>
<tr>
<td>international boundary</td>
<td>Basemap: International boundary</td>
<td>line</td>
<td>D3_international_boundaries</td>
</tr>
<tr>
<td>mainroad</td>
<td>Basemap: Main roads</td>
<td>Complex line</td>
<td>D4_main_roads</td>
</tr>
<tr>
<td>RR_basin_fao</td>
<td>Basemap: RR_basin_fao</td>
<td>Polygon Fill, Border and</td>
<td>D5_rr_basin_fao</td>
</tr>
<tr>
<td>watershed_boundaries</td>
<td>Basemap: Watershed Boundaries</td>
<td>Polygon Fill, Border and</td>
<td>D1_watershed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>transparency</td>
<td></td>
</tr>
</tbody>
</table>
Chapter 2

The Red River Simulation suite

The simulation of the system is provided using different software.

A Geomorphologic 1D mobile river bed model Simulator: it is a numerical model to solve balance equations (in conservative form) describing unsteady free surface flow in channels of complex geometry. In particular was implemented a calculation scheme to solve the mass and momentum balance of the liquid phase, the mass conservation of the streambed material, the slope friction and, finally, the sediment discharge in volume. This simulator was produced by Ferrara University, by instantiating a model of their property within a cooperation agreement that establishes that only the results are transferred to IMRR. That is why the geomorphological model code is not included in Red-TwoLe. Only experiments results are part of IMRR and described in the report D5.2.

A second Distributed Hydrologic Simulator was builded to reproduce as accurately as possible the process in the Red River delta, based on a 1D hydrodynamic model (MIKE 11). This model was set up within IMRR, by interconnecting component models previously produced by IWRP. The model, though obtained within IMRR, is not part of Red-TwoLe, since IWRP did not accept to release the configuration files. Therefore neither the software (which is protected by copyright of DHI and cannot be redistributed), nor the configuration files are made available. Only the experiments results are in report D5.3.

However, the complexity of this model makes it very slow\(^1\) thus allowing the simulation of only few user selected alternative. So, to estimate the 31 evaluation indicators under a user-selected scenario for all the efficient alternatives (thousands) it has been necessary to identify a less accurate Evaluation-oriented Simulator. It is based on a set of matlab/octave functions to reproduce the reservoirs behaviour, the Van Coc dyke system, the irrigation system and the routing in the Delta until Hanoi.

The last is the Design-oriented simulator: it is the least accurate but the

\(^1\)It takes 2 days for simulating 16 years
fastest simulator. It is implemented in C++, in order to reduce its computational requirement, as the policy design process requires to run millions of simulations. It is based on yaml, a human friendly data serialization standard for programming languages.

The following pages contain the relative manual.
Copyright

Copyright (C) 2013-2015 Emanuele Mason.

Dynamic Meta-modeling Tool is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by the Free
Software Foundation, either version 3 of the License, or (at your option) any later
version.

Dynamic Meta-modeling Tool is distributed in the hope that it will be useful, but
WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY
or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public
License for more details.

You should have received a copy of the GNU General Public License along with
Dynamic Meta-modeling Tool. If not, see http://www.gnu.org/licenses/.

Copyright of this manual

Copyright (C) 2013-2015 Emanuele Mason.

Permission is granted to copy, distribute and/or modify this document under the
terms of the GNU Free Documentation License, Version 1.3 or any later version
published by the Free Software Foundation; with no Invariant Sections, no Front-
Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section
entitled "GNU Free Documentation License".
# Contents

Copyright ii  
List of Figures vii  
Listings vii  

## 1 Gettin jiggy wit it 9

1.1 First steps ........................................ 9  
1.1.1 Function library .................................. 9  
1.1.2 Post processing .................................. 12  
1.1.3 Simulation details ................................. 12  
1.2 How to evaluate objectives .......................... 13  
1.2.1 Take advantage of the terminal .................. 14  
1.2.2 Log file ........................................ 15  
1.3 How to extract simulation data ...................... 16  
1.3.1 Simulation data format and file ................ 16  
1.4 How to add external timeseries .................... 16  
1.4.1 Create a timeserie file ........................... 17  
1.4.2 Using the timeserie file .......................... 17  

## 2 Features - building blocks 19

2.1 Functions for the simulation ........................ 19  
2.1.1 The brick: the variable .......................... 19  
2.1.2 muParser feature description .................... 20  
2.1.3 Universal approximators: ANN and RBF .......... 23  
2.2 Post processing statistics .......................... 26  
2.2.1 Steps that count ................................ 26  
2.2.2 Separable objective .............................. 27  
2.2.3 Secondary objective .............................. 28  
2.3 Simulation settings ................................ 28  
2.3.1 Horizon ........................................ 28
List of Figures

1.1 The system that will be simulated in this chapter. . . . . . . . . . . . 10

Listings

1.1 A complete functions library . . . . . . . . . . . . . . . . . . . . . . 11
2.1 A post processing statistic example . . . . . . . . . . . . . . . . . . 26
2.2 A secondary statistic example . . . . . . . . . . . . . . . . . . . . . 28
2.3 A horizon example . . . . . . . . . . . . . . . . . . . . . . . . . . . 29
1 Gettin jiggy wit it

While showing the program to fellow colleagues of mine, I figured out that is much easier to learn how to use this program step by step, even with all the up and downs that may happen, than starting with a complete list of features that will empower you to produce studies like Giuliani et al. 2015.

1.1 First steps

In this tutorial, I assume you have a computer with a working installation of DMMT. I will explain how to install the program later. Instead here we need to write the settings file to define the simulation to be performed. We want to simulate a system composed of one regulated reservoir, where the release decision $u_t$ cause a release $r_{t+1}$ from the storage $s_t$, one inflow $q_{t+1}$ and two performance measures, one measuring the excess of the lake level w.r.t. a certain threshold and the other measuring the water deficit w.r.t. a specified agricultural demand. The system is depicted in figure 1.1.

1.1.1 Function library

The core part of the settings file is the function library. The settings file is a plain text file written according to the YAML language. To write a function, you need to write the following code in a normal text editor.\(^1\)

```yaml
functionsLibrary:
  - # Reservoir L
    - Y: L:s:+1
    X: [L:s, L:q:+1, L:r:+1]
    f: "L:s + L:q:+1 - L:r:+1"
```

\(^1\)Pay attention that YAML files does not accept TAB, so set your text editor so that it puts spaces instead of tabs.
Figure 1.1: The system that will be simulated in this chapter.

The function above represent the dynamical state transition function of a fictitious reservoir with no evaporation. The Y field contains the variable that the function evaluates. The X field contains a list of input variables, in the same order as they appear in the equation. In the f field there’s a string that correspond to the function. The function string can be made of variables names, operators and spaces. No other symbols are allowed. You can find more on what can be put here by reading muParser library documentation v2.2.2, available at http://muparser.beltoforion.de/.

The variables are named according to these rules: they may be composed of three different pieces. The first is the domain’s name, like L for Lake, or IRR for the irrigation [system]. There’s an additional specific name that can be added after the colon, as in the above equation L:s or L:r for lake’s storage and lake’s release; domain and name might be any uppercase or lowercase character and underscore. Lastly, there is an optional delay value, delimited by the colon, whose sign must always be specified. If delay is not present, it is assumed to be zero. The available combination are: domain, domain:name, domain:delay, domain:name:delay. I don’t remember what happens without the domain, so just assume it is undefined behavior.\(^2\)

The listing 1.1 shows a complete functions library.

\(^2\)To be more precise, you can use 0123456789_-a ... zA ... Z to build the name. Also the first character should be a letter.
Listing 1.1: A complete functions library

```python
functionsLibrary:

# Reservoir L
- Y: L:s:+1
  X: [L:s, L:q:+1, L:r:+1]
  f: "L:s + L:q:+1 - L:r:+1"

- Y: L:r:+1
  X: [L:s, L:u]
  f: "max( max( L:s - 100, 0 ), min( L:s, L:u ) )"

- Y: L:q:+1
  X: []
  f: "40"

- Y: L:u
  X: [L:s]
  f: RBF
  architecture: 3
  uniformization: scaling
  uniformizers:
    - name: L:u
      values: [0, 155]
    - name: L:s
      values: [0, 155]

# stepcosts
- Y: L:g1:+1
  X: [L:s:+1]
  f: "max( L:s:+1 - 50, 0 )"

- Y: IRR:g2:+1
  X: [L:r]
  f: "max( 50 - L:r, 0 )"
```

With the code above, we have defined the state transition function of the system, a constant value for the input L:q:+1 and the class of function of the control rule L:u i.e., a class of universal approximators called radial basis function. Under the section stepcosts we have defined the functions for the exit transformations.
1.1.2 Post processing

What we need from the simulation are summary statistics representative of system performance, namely average value of the stepcosts. In our system we have four of them. We define them with the following code.

```python
postprocessing:
    # flooding on the L
    - name: A_L:mean_daily_h_excess
      type: SeparableObjective
      dataRequired: [L:g1]

    # irrigation
    - name: B_IRR:mean_daily_deficit
      type: SeparableObjective
      dataRequired: [IRR:g2]
```

I know, life can be though sometime. The printing order in the output is defined by the alphabetical ordering of the names, but to make it easier we usually put a letter in front of the name. An order: field may be available in future releases.

1.1.3 Simulation details

Now we need to define some additional information that will define the simulation. The first one is what the program is supposed to do in a single step of the simulation. We need the following code. It defines the list of the input variable X for the whole step evaluation, which can be retrieved from previous step evaluation or from an external file. In the field parameters there's a variable which represent the metafunction whose parameters should be defined during program execution. Finally, in the field Y there is a list of target variables to be evaluated.³

```python
simulation:
    singleStep:
        parameters: [L:u]
        X: [L:s]
```

³This feature will be removed in future because it can be deducted from the postprocessing: and/or the simulationVarsToLog: sections.
# THESE ARE THE TARGET VARIABLES THAT TRIGGERS THE EVALUATION CHAIN AT EACH STEP.
Y: [L:s:+1, L:g1:+1, IRR:g2:+1, HYD:g3:+1, FLO:g4:+1]

The second one is the horizon: how many steps of simulation are needed. Right now the definition is quite error-prone: it requires the user to specify the step that represents \( t = 0 \) in the simulation and the one that represents \( t = H \). The required YAML code will look like the following.

```yaml
horizon:
  length: 101 # 0 based, enough to contain all the simulation data
  firstStep: 0 # 0 based
  lastStep: 100 # 0 based, last step of evaluation of "singleStep: Y: [stuff]"
```

The last you thing you may need is the initialization section, in which you can put single values of variables that cannot be evaluated within the simulation.

```yaml
initialization:
  - name: L:s
    values: # [at the step #X, value]
      - [0, 100]
```

## 1.2 How to evaluate objectives

Now that we have defined the full system within our YAML settings file, we are now ready to evaluate the value of the objectives, a.k.a. postprocessing statistics. To do that, you just need to open a unix terminal and launch the following command, having named the settings file `lake.yaml`.

```
my-laptop:~ lordmzn$ ./DMMT -s lake.yaml
```

To open a unix terminal on your computer: search in Google for "open terminal <your operating system>".
Otherwise you can also use the long version of the command line option.

```
my-laptop:~ lordmzn$ ./DMMT --settings lake.yaml
```

If everything is working, nothing should happen, as in every GNU-like app. The program is waiting for you to type in the value of the parameters you want to simulate. These parameter values will become the parameter of the function that evaluates the variables contained in the U: field in simulation: singleStep: section. So just type in the values, separated by any usual delimiter (spaces or tabulation are ok), then press return/enter. This will trigger a simulation, and the program will print in a new line the values of the corresponding objectives we requested. Here’s an example.\(^5\)

```
my-laptop:~ lordmzn$ ./DMMT -s lake.yaml
0.1 0.5 0.7 0.6 0.3 0.1 0.5 0.4
0.5574961478471852 -2
```

You can then go on by inserting other values, to obtain other objectives values. When you’ve finished, just insert a newline without typing any additional character, and the program will halt.

### 1.2.1 Take advantage of the terminal

Working with standard input and output let us do some very nice tricks with the terminal stream redirection capabilities. We can save the objective values in a file instead of printing them to the screen.\(^6\)

```
my-laptop:~ lordmzn$ ./DMMT -s lake.yaml > obj.txt
0.1 .2 .4 .6 .8 .4 .5 .1 .3
0.5 .7 .2 .6 .7 .3 .1 .5 .4
my-laptop:~ lordmzn$ cat obj.txt
0.26378941136176781 -2
```

\(^5\)You can even add more variables to a single line of parameters, only the first ones will be read and evaluated.

\(^6\)cat is a GNU program that display the contents of a file as a text file.

\(^7\)You can notice the short syntax for decimal, without the leading zero. And also the last .4 that is silenty ignored.
1.2 HOW TO EVALUATE OBJECTIVES

We can redirect the input, so that we simulate the parameters values contained in a file.

```
1 my-laptop:~ lordmzn$ cat ctrl.txt
2 .1 .2 .4 .6 .8 .4 .5 .1 .3
3 .1 .5 .7 .2 .6 .7 .3 .1 .5 .4
4 my-laptop:~ lordmzn$ ./DMMT -s lake.yaml < ctrl.txt
5 0.26378941136176781 -2
6 0.55574961478471852 -2
```

You can even do both, so that the terminal output remains clean.

```
1 my-laptop:~ lordmzn$ ./DMMT -s lake.yaml < ctrl.txt > obj.txt
2 my-laptop:~ lordmzn$ cat obj.txt
3 0.26378941136176781 -2
4 0.55574961478471852 -2
```

Unless any error appears, of course.

```
1 my-laptop:~ lordmzn$ cat ctrl.txt
2 .1 .2 .4 .6 .8 .4 .5 .1 .3
3 .1 .6 .7 .3 .1 .5 .4
4 my-laptop:~ lordmzn$ ./DMMT -s lake.yaml < ctrl.txt > obj.txt
5 Unable to parse double variable
6 my-laptop:~ lordmzn$ cat obj.txt
7 0.26378941136176781 -2
```

1.2.2 Log file

If you're actually trying the above commands, you will notice that a new file will pop up. It's a normal text file that contains useful information collected during the simulation. The level of detail can be set at compile time for now, but in the future it should be revealed with a command line option. For now, you just need to know that if everything is working correctly, the file should be empty.
You can specify a name for the file via command line option. The name of the option is `LOGFILE` or just `L` in the short version, followed by the name of the file.

### 1.3 How to extract simulation data

To extract simulation data, you will need an additional field in the `SIMULATION:` section, like the following. The variable names within the list have to be evaluable in someway and their values will be collected for each step of the simulation:\(^8\)

```
1 simulationVarsToLog: [L:s, L:u, L:r:+1, L:HP:+1, L:g1:+1, ←
IRR:g2:+1, HYD:g3:+1, FLO:g4:+1]
```

Having added that to our `LAKE.YAML`, we can then launch the program with a command like the following.

```
1 my-laptop:~ lordmzn$ cat controls
2 .9 .2 .4 .6 .8 .4 .5 .1 .3
3 .9 .1 .6 .7 .3 .1 .5 .4
4 my-laptop:~ lordmzn$ ./DMMT --sett lake.yaml --logFile log.log ←
   --simFile sim < ctrls.txt > obj.txt
```

This will produce two files named `sim_0.log` and `sim_1.log` containing the values for each step of the simulation. You will notice that this reservoir is quite stable and even a monkey can control it, since after a few steps it's already at equilibrium.

#### 1.3.1 Simulation data format and file

Remember that in each line you will find all the variables of the same simulation step.

### 1.4 How to add external timeseries

As we have seen now, the system reaches its equilibrium point very quickly. This is mainly due to the fact that the inflow is constant, which is a rather unrealistic

\(^8\)Otherwise you will get a column of -2.
1.4 HOW TO ADD EXTERNAL TIMESERIES

hypothesis. We can create a synthetic timeseries of inflows and simulate the system under the new driver and see what happens.

1.4.1 Create a timeserie file

To do that, open Matlab®, change folder to the same folder as of DMMT and write the following code.

```matlab
1   >> q_t = randn(101,1) * 10 + 40;
2   >> save timeserie.txt q_t -ascii
```

Now open timeserie.txt with any text editor and add this line at the top: /*:<L:q>/:** it will identify the data in the column as a realization of variable L:q.⁹

1.4.2 Using the timeserie file

Before trying the simulation with the new data, you should modify the settings file accordingly: you should remove the definition of the constant L:q:+1 and instead add it to the list of simulation: singleStep: X: variables.

To perform simulations with the newly added timeserie.txt, you should use a command like the following.

```bash
1   my-laptop:~ lordmzn$ ./DMMT --sett lake.yaml --logFile log.log ←
2       --simFile sim --inputFile timeserie.txt < ctrls.txt > obj.txt
```

⁹You can use this file format to load also matrices into DMMT, given that each line represent the same simulation step and the first line contains corresponding names.
2 Features - building blocks

Three parts join together to define the simulation engine called DMMT: the functions defining what is evaluated at each step of the simulation, the statistics that should be evaluated after the simulation as a recap of the full simulation and finally, the settings of the simulation itself - length, which variables should be evaluated and so on.

The parts correspond to three different fields in the YAML settings file. We can look at these fields as section of the settings file, with the field names, the keys, serving as section titles. They are: functionsLibrary, postprocessing, simulation.

2.1 Functions for the simulation

The function library is defined by the key functionsLibrary; the library itself is a list of functions. Each function has three mandatory fields, Y, X and f.

Y can be a sequence of variables or just one. It represents the output of the function.

X is a sequence of variables representing the inputs to the function.

f the function itself. Must be a string but can be of different types:

"muParser" Matlab-like analytical functions;
"ANN" Artificial Neural Networks;
"RBF" Radial Basis Functions networks.

2.1.1 The brick: the variable

The basic element of the YAML setting file is the variable definition. The variables names may be composed of three different pieces.

• The first is the domain name, like L for Lake, or IRR for the irrigation (system).
• There’s an additional specific name that can be added after the colon, as in L:s or L:r for lake storage and lake release; domain and name might be any uppercase or lowercase character and underscore.

• Lastly, there is an optional delay value, delimited by the colon, whose sign must always be specified. If delay is not present, it is assumed to be zero.

So the available combination are: A, A:x, A:x, A:x:+1. I don’t remember what happens without the domain, so just assume it is undefined behaviour. To be more precise, you can use 0123456789:_:+-a ... zA ... Z to build the name. Also the first character should be a letter.

### 2.1.2 muParser feature description

MuParser is the name of the library that provides the functionality to transform the text contained in the YAML settings file into a function object that can be used by DMMT. Therefore the syntax of this class is closely resembling the one of the muParser library.

More specifically, a muParser function is a multiple-input single-output (MISO) function that is specified in YAML with just three fields:

- **Y** is the output. It can be one a scalar or a sequence with just one element - the output variable.

- **X** is a sequence of variables representing the inputs to the function.

- **f** is the function itself. It must be a string, so we strongly suggest you to surround it with "" so the YAML parser does not get confused by any other symbol. It contains only constant numbers or symbols representing muParser operators or variables taken from the list above. The allowed symbols can represent muParser built-in functions, binary operators or the only ternary operator.

The table 2.1 gives an overview of the functions supported by muParser. It lists the function names, the number of arguments and a brief description. The supported built-in binary operators are listed in table 2.2. MuParser has only one built-in ternary operator right now (table 2.3), that gives support for the if-then-else operator. It uses lazy evaluation in order to make sure only the necessary branch of the expression is evaluated.

An example of muParser function is given in listing below.
2.1 Functions for the Simulation

```
1 - Y: HN:q
2  X: [VT:q, HN:h]
3  f: "VT:q - (HN:h > 13.4 ? min( max( 4166.7 * ((HN:h - 12.8), 0 ), 2500 ) : 0)"
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Argc.</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>sin</td>
<td>1</td>
<td>sine function</td>
</tr>
<tr>
<td>cos</td>
<td>1</td>
<td>cosine function</td>
</tr>
<tr>
<td>tan</td>
<td>1</td>
<td>tangens function</td>
</tr>
<tr>
<td>asin</td>
<td>1</td>
<td>arcus sine function</td>
</tr>
<tr>
<td>acos</td>
<td>1</td>
<td>arcus cosine function</td>
</tr>
<tr>
<td>atan</td>
<td>1</td>
<td>arcus tangens function</td>
</tr>
<tr>
<td>sinh</td>
<td>1</td>
<td>hyperbolic sine function</td>
</tr>
<tr>
<td>cosh</td>
<td>1</td>
<td>hyperbolic cosine</td>
</tr>
<tr>
<td>tanh</td>
<td>1</td>
<td>hyperbolic tangens function</td>
</tr>
<tr>
<td>asinh</td>
<td>1</td>
<td>hyperbolic arcus sine function</td>
</tr>
<tr>
<td>acosh</td>
<td>1</td>
<td>hyperbolic arcus tangens function</td>
</tr>
<tr>
<td>atanh</td>
<td>1</td>
<td>hyperbolic arcur tangens function</td>
</tr>
<tr>
<td>log2</td>
<td>1</td>
<td>logarithm to the base 2</td>
</tr>
<tr>
<td>log10</td>
<td>1</td>
<td>logarithm to the base 10</td>
</tr>
<tr>
<td>log</td>
<td>1</td>
<td>logarithm to the base 10</td>
</tr>
<tr>
<td>ln</td>
<td>1</td>
<td>logarithm to base e (2.71828...)</td>
</tr>
<tr>
<td>exp</td>
<td>1</td>
<td>e raised to the power of x</td>
</tr>
<tr>
<td>sqrt</td>
<td>1</td>
<td>square root of a value</td>
</tr>
<tr>
<td>sign</td>
<td>1</td>
<td>sign function -1 if x&lt;0; 1 if x&gt;0</td>
</tr>
<tr>
<td>rint</td>
<td>1</td>
<td>round to nearest integer</td>
</tr>
<tr>
<td>abs</td>
<td>1</td>
<td>absolute value</td>
</tr>
<tr>
<td>min</td>
<td>var.</td>
<td>min of all arguments</td>
</tr>
<tr>
<td>max</td>
<td>var.</td>
<td>max of all arguments</td>
</tr>
<tr>
<td>sum</td>
<td>var.</td>
<td>sum of all arguments</td>
</tr>
<tr>
<td>avg</td>
<td>var.</td>
<td>mean value of all arguments</td>
</tr>
</tbody>
</table>

Table 2.1: Built-in functions in the muParser library.
<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
<td>assignment</td>
<td>-1</td>
</tr>
<tr>
<td>&amp;&amp;</td>
<td>logical AND</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;=</td>
<td>less or equal</td>
<td>4</td>
</tr>
<tr>
<td>&gt;=</td>
<td>greater or equal</td>
<td>4</td>
</tr>
<tr>
<td>!=</td>
<td>not equal</td>
<td>4</td>
</tr>
<tr>
<td>==</td>
<td>equal</td>
<td>4</td>
</tr>
<tr>
<td>&gt;</td>
<td>greater than</td>
<td>4</td>
</tr>
<tr>
<td>&lt;</td>
<td>less than</td>
<td>4</td>
</tr>
<tr>
<td>+</td>
<td>addition</td>
<td>5</td>
</tr>
<tr>
<td>-</td>
<td>subtraction</td>
<td>5</td>
</tr>
<tr>
<td>*</td>
<td>multiplication</td>
<td>6</td>
</tr>
<tr>
<td>/</td>
<td>division</td>
<td>6</td>
</tr>
<tr>
<td>^</td>
<td>raise x to the power of y</td>
<td>7</td>
</tr>
</tbody>
</table>

*The assignment operator is special since it changes one of its arguments and can only be applied to variables.

Table 2.2: Built-in binary operators in the muParser library.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>?:</td>
<td>if then else operator</td>
<td>C++ style syntax</td>
</tr>
</tbody>
</table>

Table 2.3: Built-in ternary operators in the muParser library.
2.1.3 Universal approximators: ANN and RBF

Sometimes, the adoption of universal approximators is required to provide more flexibility to the shape of a certain function. In this program, we consider two universal approximators to define the parameterized universal approximators function class: Artificial Neural Networks and gaussian Radial Basis Functions.

Both the two classes of functions can be used with some common YAML settings, an example of which is given below.

```
- Y: [A:y, B:y]
- X: [A:x1, A:x2, B:x]
  f: ANN
  architecture: 7
  parameters: [0.99, 0.53, -0.68, 0.49, 0.23, 0.14, ...]
  uniformization: scaling
  uniformizers:
    - name: A:y
      values: [0, 100]
    - name: A:x1
      values: [0, 1]
```

A field by field description is provided.

- **Y** can be a sequence or a scalar, since ANN and RBF are multiple-input multiple-output (MIMO) functions.
- **X** must be provided as a non-empty sequence.
- **f** can have only two possible values so far:
  - “ANN” for Artificial Neural Networks;
  - “RBF” for Radial Basis Functions.
- **architecture** is the number of component of the approximator - can be neurons or basis functions, according to the specific function class. It's required.
- **parameters** is a vector of parameters that must be specified to identify a specific function already calibrated. The parameters can be set up before starting the program, or at runtime if the field `singlesStep: parameters:` contains the output of the function. To set them at runtime, you can either write the
values in a file pointing it with the -n command line option or type them into the stdin.\textsuperscript{1} In both these cases, any value specified in the field parameters of this function will be overwritten (in fact you may omit it). If instead you don’t want to set the parameters at runtime, you have to provide a vector of values in the \textit{parameters} field. Order and number of parameters depends on the specific function class.

\textit{Uniformization} specifies the preprocessing treatment of the inputs to this function and the postprocessing of the outputs too. It’s an optional value. If you omit it, it will take the value \textit{none}. You can choose between three options:

- \textbf{Scaling} rescales the input variable $x$ uniformly between two specified values, a minimum and a maximum. Performs the opposite on the output variable $y$, as in equation 2.1.

$$f_{\text{input}} = \frac{x - \text{min}}{\text{max} - \text{min}}$$

$$y = \text{min} + f_{\text{output}}(\text{max} - \text{min})$$

(2.1)

- \textbf{Gaussianization}, also called normalization, subtracts to each input its average value and then divides the result for its standard deviation. For an output, it multiplies it for the standard deviation and then adds the average, as in equation 2.2. Works well for data distributed as a normal variable.

$$f_{\text{input}} = \frac{x - \text{avg}}{\text{std}}$$

$$y = \text{avg} + f_{\text{output}}(\text{std})$$

(2.2)

- \textbf{None} does nothing. It’s the default value.

\textit{Uniformizers} is a sequence of parameters for the uniformization. If you have specified a uniformizer other than none, it’s a mandatory field. The sequence is composed of two-field elements: the first is name and should correspond to either an input variable or an output variable. The second is values and must be a vector containing two elements:

- $\text{min}$, $\text{max}$ in case of scaling.
- $\text{avg}$, $\text{std}$ in case of gaussianization.

\textsuperscript{1}\textit{Standard input stream}, which is usually what is typed on the keyboard but can be substituted with a file via the redirection operators of the terminal, $<$. 

24
Let’s now have a look into the two specific classes of universal approximator functions.

2.1.3.1 Artificial Neural Networks

An artificial neural network is a multiple-input multiple-output (MIMO or vector) function where the \( k \)-th component in the output vector \( y_t \) (with \( k = 1, \ldots, K \)) is defined as:

\[
y^k_t = a_k + \sum_{i=1}^{N} b_{i,k} \psi_i(\chi_t \cdot c_{i,k} + d_{i,k})
\]  
(2.3)

where \( N \) is the number of neurons \( \psi(\cdot) \) (i.e., the number of hyperbolic tangent sigmoid functions that ensure universal approximation properties), \( \chi_t \in \mathbb{R}^M \) the function input vector, \( a_k, b_{i,k}, d_{i,k} \in \mathbb{R}, c_{i,k} \in \mathbb{R}^M \) the ANN parameters.

The parameter vector \( \theta \) belongs to \( \mathbb{R}^{n_\theta} \), where \( n_\theta = K(N(M+2)+1) \) and is defined as:

\[
\theta = [\ldots d_{1,1} \ldots d_{N,1}] a_1 [b_{1,1} \ldots b_{N,1}] [c_{1,1,1} \ldots c_{1,M,1} \ldots c_{N,1,1} \ldots c_{N,M,1}] \ldots
\]

\[
[[d_{1,K} \ldots d_{N,K}] a_K [b_{1,K} \ldots b_{N,K}] [c_{1,1,K} \ldots c_{N,1,K} \ldots c_{1,M,K} \ldots c_{N,M,K}]],
\]  
(2.4)

with \( N \) is again the number of neurons, \( K \) is the dimension of the output vector and \( M \) the dimension of the input one.

2.1.3.2 Radial Basis Functions

A radial basis function (network) is a MIMO (vector) function where the \( k \)-th output variable in the vector \( y_t \) (with \( k = 1, \ldots, n_y \)) is defined as:

\[
y^k_t = \sum_{i=1}^{N} w_{i,k} \varphi_i(\chi_t)
\]  
(2.5)

where \( N \) is the number of RBFs \( \varphi(\cdot) \) and \( w_{i,k} \) the weight of the \( i \)-th RBF. The weights are formulated such that they sum to one (i.e., \( \sum_{i=1}^{N} w_{i,k} = 1 \)) and are non-negative (i.e., \( w_{i,k} \geq 0 \quad \forall i, k \)). The single RBF is defined as follows:

\[
\varphi_i(\chi_t) = \exp \left[ -\sum_{j=1}^{M} \frac{(\chi_t)_j - (c_{j,i})^2}{(b_{j,i})^2} \right]
\]  
(2.6)
where $M$ is the number of input variables $\chi_t$ and $c_i, b_i$ are the $M$-dimensional center and radius vectors of the $i$-th RBF, respectively. The centers of the RBF must lie within the bounded input space and the radii must strictly be positive (i.e., using normalized variables, $c_i \in [-1, 1]$ and $b_i \in (0, 1]$).

The parameter vector $\theta$ belongs to $\mathbb{R}^{n_\theta}$, where $n_\theta = N(2M + K)$ and is defined as:

$$
\theta = [c_1 r_{1,1} \ldots c_M r_{M,1} \quad w_{1,1} \ldots w_{K,1}] 
\ldots
\quad [c_1 r_{1,N} \ldots c_M r_{M,N} \quad w_{1,N} \ldots w_{K,N}],
$$

with $N$ is again the number of neurons, $K$ is the dimension of the output vector and $M$ the dimension of the input one.

### 2.2 Post processing statistics

The post processing part is defined by the key postprocessing; it’s a sequence of specific objects. The objects are called objective because they represent the entire simulation in a single value or in a vector of values. The thing they have in common with statistics is that they are vector functions: they take vector as inputs and give a scalar as output.

They all accept some fields, shown below. Among them, only name and type are mandatory.

Listing 2.1: A post processing statistic example

```plaintext
- name: ANY:variable_name
  type: SeparableObjective
  stepsThatCount: [1, 10]
  dataRequired: [A:variable]
```

There are two available type of statistics/objectives: separable and secondary. Before entering into details, we should explain the stepsThatCount.

#### 2.2.1 Steps that count

Given the variety of events and situations that can be simulated within the a single run of the model, there is the need to focus the assessment of a specific statistic on just
some of the events or situations. This is the purpose of the field `stepsThatCount`: it contains references to the steps that counts toward the evaluation of a certain statistic.

How it works? You should know that the C language heritage of DMMT means that the indexes of the steps are 0 based so the first step is the number 0. Given that, the steps are indicated via ranges:

- if the vector `stepsThatCount` is empty, we have the default behaviour which means that any step counts;
- if the vector has just one element, the steps that counts are the ones from the absolute first step up to the specified step, than none more;
- if it has two values, it will counts steps from the first index specified up to the second index;
- if it has three values, it will counts steps from the absolute first step up to the first index, then from the second index to the third;
- and so on and so forth.

### 2.2.2 Separable objective

A separable objective is the simplest statistics you can think of. It takes one variable as `dataRequired`, retrieving all the values of that variable with timesteps within the ranges of `stepsThatCount`. It evaluates the average value or the maximum of these data.

Remember to not put any delay in the `dataRequired` field; if you need `L::g1::+1`, put `L::g1` in the `dataRequired` field. You should instead specify the absolute steps you want in `stepsThatCount`.

For this kind of statistics, it is mandatory to have one and only one `dataRequired`; the `stepsThatCount` field can be omitted, in which case all the simulation steps count. The aggregator fields let the user to specify the desired statistics, namely average or max - the field can be omitted and average will be used.

An example is given in the listing 2.1 at the start of this section.
2.2.3 Secondary objective

A secondary objective is an objective that depends on one or more objectives - like a variance. It’s not related to steps, and therefore it does not accept any stepsThatCount. Instead, it resembles a muParser function (subsection 2.1.2), with dataRequired used as the sequence of inputs and a mandatory field f with the string that represent the formula. An example is given in the listing 2.2.

Listing 2.2: A secondary statistic example

```
1 - name: ANY:variable_name
2    type: SecondaryObjective
3    dataRequired: [A:average, A:average_of_squares]
4    f: "A:average_of_squares - (A:average^2)"
```

2.3 Simulation settings

The settings of the simulation are defined in the YAML file in the field simulation. There are four subparts:

- horizon;
- parameter;
- singleStep;
- initialization;
- simulationVarsToLog.

2.3.1 Horizon

This section specifies the number of steps of the simulation. Specifically, each field can have a scalar unsigned integer value. As for the stepsThatCount field, also here the indexes are 0 based.

The entire field is not mandatory: if absent, it will default to a 2-step simulation. If instead you put it in, a description of the available options is given below.
2.3 simulation settings

length is the number of different simulation steps that the program will create;

firstStep is the index of the step at which the evaluation of the variables will start — the default is 0;

lastStep is the index of the last step at which the evaluation of the variables is performed — the default is length - 1;

rowsToSkipInDataFile refers to the number of rows the program should skip in the external input file — the default is 0.

An example of this specification is given in the code snippet 2.3.

Listing 2.3: A horizon example

1 simulation:
2   horizon:
3       length: 102
4       firstStep: 1
5       lastStep: 100
6       rowsToSkipInDataFile: 12

2.3.2 Parameters

This field defines the runtime parameters of the simulation, if any. The field is optional: without it, once the DMMT executable is started, it can produce only one specific simulation (but right now, it will segfault because of a bug). To define a runtime parameter, there should be a “parameterizable” function in the functions library. In that case, one of its output can be put in the sequence parameters. Then, at runtime, values for those parameters can be changed, for example via stdin or via a parameter file with the -n command-line option. The order of the parameters depends on the specific function targeted.

Soon, it will be possible to change parameters of more than one function by simply adding more variables to this sequence. Than the order of parameters will be the composition of the order of the specific function parameters according to the order of the variables in the sequence.

An example of this specification is given in the code snippet below.
2.3.3 Single step

This field serves to define what happens in the generic step. It has two subfields: X: contains the variable to start the step evaluation from; Y: contains the target variables that must be evaluated at each step. The program is capable to generate the tree of the functions to be evaluated in order to calculate a certain target variable. This means that here you just have to put the ones that aren’t needed by any other (usually: exit transformations) while you don’t have to put all the variables that should be evaluated within a step.

An example of these specifications is given in the code snippet below.

```
simulation:
  parameters: [L:u]

singleStep:
  X: [L:s]
  Y: [L:g1:+1, IRR:g2:+1]
```

2.3.4 Initialization

This section provides initial values for the variables that can then be evaluated through simulation. Which variables are these? They’re state variables which dynamic behaviour is modeled by a function; they’re variables with negative delay, large enough that the model should know their value to start the evaluation of the first step.

An empiric rule is the following: from the sequence in `singleStep: X:`, you can exclude the one provided with the input data file. All the remaining must have a value to start from inserted in this section.

In the field, there should be a sequence of elements. Each element must have two fields: name: and values:. name is a scalar and represent a variable. values is a sequence of elements itself. Each element is a vector with one integer and one real number: the first is the absolute index of the simulation step (0-based), the second is the initial value that the program will give to the variable name at that step.

An example of this specification is given in the code snippet below.
2.3 Simulation settings

This field is required only when the program is launched with the `-f FILENAME` option. It contains a sequence of variables that one wants to log. The delay here can be used to shift the variable of one or more line within the log file.

For each simulation executed, a log file will be produced. They will have a number at the end of the filename, indicating a sequential index of the simulations performed during a single program execution.

An example of this specification is given in the code snippet below.

```
simulation:
  initialization:
    - name: `L:s`
      values: `# [absoluteStepID, value]`
      - `[0, 100]`
```

2.3.5 Simulation variables to log

This field is required only when the program is launched with the `-f FILENAME` option. It contains a sequence of variables that one wants to log. The delay here can be used to shift the variable of one or more line within the log file.

For each simulation executed, a log file will be produced. They will have a number at the end of the filename, indicating a sequential index of the simulations performed during a single program execution.

An example of this specification is given in the code snippet below.

```
simulation:
  simulationVarsToLog: `[L:s, L:u, L:r:+1, L:HP:+1, L:g1:+1]`
```
Bibliography

Giuliani, M., A. Castelletti, F. Pianosi, E. Mason, and P.M. Reed (2015). “Curses, tradeoffs, and scalable management: advancing evolutionary multi-objective direct policy search to improve water reservoir operations”. In: Journal of Water Resources Planning and Management (cit. on p. 9).

A Writing this software

We have seen that computer programming is an art, because it applies accumulated knowledge to the world, because it requires skill and ingenuity, and especially because it produces objects of beauty.

— Knuth, “Computer Programming as an Art”, 1974

A.1 How to write good software: evaluating performances

The program relies on a certain amount of data which must be produced and held during the execution of the software. An optimization run usually lasts for tens of hours and therefore the memory management becomes an important issue. Assessment of its correctness and coherence has been performed using Valgrind.

Valgrind is an instrumentation framework for building dynamic analysis tools. There are Valgrind tools that can automatically detect many memory management and threading bugs, and profile your programs in detail. It runs on the following platforms: X86/Linux, […]. Valgrind is Open Source / Free Software, and is freely available under the GNU General Public License, version 2.

Performance is also a critical issue: during the optimization, the operations performed by the model are executed millions of times. Therefore we analyzed performance in two ways. A first analysis have been performed during development with a profiler, gprof. Profiling allows you to learn where the program spent its time and which functions called which other functions while it was executing. This information can show which pieces of your program are slower than expected and might be candidates for rewriting to make the program execute faster.

The program however reached soon a level of complexity that gprof wasn’t capable of handling. Therefore we switched to a different profiling paradigm which is much
more efficient while still providing sound proofs of effectiveness. An informal but
detailed explanation is given here: http://stackoverflow.com/questions/
375913/what-can-i-use-to-profile-c-code-in-linux. Briefly, the idea
is to run the program in a debugger that allows you to stop it randomly sometime and
to look into the stack where you can see the function it was executing. Repeating
the process even a moderate number of times leads to very interesting insights. In
our case, the class representing the variable itself have been deeply optimized after
this techniques demonstrated that early run were spent almost completely running its
inefficient code. If one has ever used gprof on a large program and hates the way
profiling data are presented or how long and annoying is the profiling process, we
strongly suggest to give a try to this method.

A.2 Unsung heroes

A list of the open source resources that made this program possible is given here.

A.2.1 Developing C++ code

- YAMLcpp: https://github.com/jbeder/yaml-cpp
- muParser: http://muparser.beltoforion.de or https://github.com/
  beltoforion/muparser/
- boost: http://www.boost.org
  pgno=1
- to parse options: http://optionparser.sourceforge.net/index.html
- style of the code: http://google-styleguide.googlecode.com/svn/
  trunk/cppguide.html
- C++ programming resources: en.cppreference.com (please DON'T USE
  cplusplus.com)
- general questions: stackoverflow.com
A.2.2 Programming environment

- Eclipse + CDT for integrated code completion system
- sublime text for lightweight code editor
- git with BitBucket GUI
- gitlab.com for private hosting
- filezilla to move things from local environment to computing server

A.2.3 Additional work

I was asked to use the \texttt{\LaTeX} system for this manual, and I would have chosen it anyway. As the website http://latex-project.org/ states,

\texttt{\LaTeX} is a document preparation system for high-quality typesetting. It is most often used for medium-to-large technical or scientific documents but it can be used for almost any form of publishing. \texttt{\LaTeX} is not a word processor! Instead, \texttt{\LaTeX} encourages authors not to worry too much about the appearance of their documents but to concentrate on getting the right content. [It] is based on the idea that it is better to leave document design to document designers, and to let authors get on with writing documents. \texttt{\LaTeX} is based on Donald E. Knuth’s \texttt{\TeX} typesetting language or certain extensions. \texttt{\LaTeX} was first developed in 1985 by Leslie Lamport, and is now being maintained and developed by the \texttt{\LaTeX}3 Project.
B Error messages

Unable to parse double variable happens when the values sent to the evaluation are not numbers, for example

```
1 my-laptop:~ lordmzn$ ./Model -s lake.yaml
2 0.1 0.2 0.4 0.6 0.8 0.4 0.5 y
3 Unable to parse double variable
```

Attempted to parse variable but at end-of-line happens when the number of values sent to the evaluation are not enough to completely identify the function that evaluates the $U$: variables.
C Debug

C.1 Compile with increased verbosity

You can recompile the program with different levels of verbose intensity. To do that, you should define the appropriate value of the c++ define constant “FILELOG_MAX_LEVEL”. You can define it in the compilation command, as in g++ -DFILELOG_MAX_LEVEL="utilities::Lo,
The log file will have all the messages up to that level of verbosity.

The levels in order of decreasing detail are:

1. -DFILELOG_MAX_LEVEL="utilities::LogLevel::kDebug4"
2. -DFILELOG_MAX_LEVEL="utilities::LogLevel::kDebug3"
3. -DFILELOG_MAX_LEVEL="utilities::LogLevel::kDebug2"
4. -DFILELOG_MAX_LEVEL="utilities::LogLevel::kDebug1"
5. -DFILELOG_MAX_LEVEL="utilities::LogLevel::kDebug"
6. -DFILELOG_MAX_LEVEL="utilities::LogLevel::kInfo"
7. -DFILELOG_MAX_LEVEL="utilities::LogLevel::kWarning"
8. -DFILELOG_MAX_LEVEL="utilities::LogLevel::kError"

C.2 YAML correctness

D GNU Free Documentation License

Version 1.3, 3 November 2008

$http://fsf.org/$

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited
AP\textsc{pendix}

in duration, to use that work under the conditions stated herein. The “\textbf{Document}”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “\textbf{you}”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “\textbf{Modified Version}” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “\textbf{Secondary Section}” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “\textbf{Invariant Sections}” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “\textbf{Cover Texts}” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “\textbf{Transparent}” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “\textbf{Opaque}”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, \LaTeX{} input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed
for human modification. Examples of transparent image formats include PNG, XCF
and JPG. Opaque formats include proprietary formats that can be read and edited only
by proprietary word processors, SGML or XML for which the DTD and/or processing
tools are not generally available, and the machine-generated HTML, PostScript or
PDF produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following
pages as are needed to hold, legibly, the material this License requires to appear
in the title page. For works in formats which do not have any title page as such,
“Title Page” means the text near the most prominent appearance of the work’s title,
preceding the beginning of the body of the text.

The “publisher” means any person or entity that distributes copies of the Document
to the public.

A section “Entitled XYZ” means a named subunit of the Document whose title ei-
ther is precisely XYZ or contains XYZ in parentheses following text that translates XYZ
in another language. (Here XYZ stands for a specific section name mentioned below,
such as “Acknowledgements”, “Dedications”, “Endorsements”, or “History”.) To
“Preserve the Title” of such a section when you modify the Document means that it
remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states
that this License applies to the Document. These Warranty Disclaimers are consid-
ered to be included by reference in this License, but only as regards disclaiming
warranties: any other implication that these Warranty Disclaimers may have is void
and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or
noncommercially, provided that this License, the copyright notices, and the license
notice saying this License applies to the Document are reproduced in all copies, and
that you add no other conditions whatsoever to those of this License. You may not
use technical measures to obstruct or control the reading or further copying of the
copies you make or distribute. However, you may accept compensation in exchange
for copies. If you distribute a large enough number of copies you must also follow
the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may
publicly display copies.

3. COPYING IN QUANTITY
If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document’s license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document’s license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled “History”, Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled “History” in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the “History” section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled “Acknowledgements” or “Dedications”, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
APPENDIX

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents,
unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the
8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights
have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.
Chapter 3

The Red River
Optimization suite

In the IMRR project all regulation alternatives are designed using two optimization engines: the *MOEA Framework*, which was employed for the design of the operating policies for Son La, Hoa Binh, and Tuyen Quang reservoirs via Evolutionary Multi-Objective Direct Policy Search, and the *TwoLe Planning Engine*, used for the optimal operations of Thac Ba.

The MOEA Framework is a free and open source Java library for Multi-Objective Evolutionary Optimization that can be combined with the Design-oriented simulator for designing Pareto optimal operating policies. The architectural choice of separating MOEA Framework and the simulator represents a significant improvement with respect to the common monolithic optimization tools, where the system model and the optimization routine are packed together. In fact, the simulator can be easily and independently updated (for example by adding a new system components such as a new reservoir) without requiring to modify the optimization engine.

All documentation and the framework itself can be found at [http://moeaframework.org](http://moeaframework.org).

The TwoLe Planning Engine provides an alternative optimization engine relying on Stochastic Dynamic Programming, which can be used for designing optimal operating policies only in small/medium systems. For this reason, this engine cannot scale to the entire RTBR system, while it was suitable for the design of the optimal operating policy for Thac Ba reservoir.

All documentation and the program itself can be found at [http://savannah.nongnu.org/projects/twole-plan](http://savannah.nongnu.org/projects/twole-plan).
Chapter 4

Other stuff

As IMRR project result other ancillary software was developed.

To store timeseries, either data and metadata, a specific structure, namely "kali", and specific I/O interface to link all module of Red-TwoLe software was implemented in octave/matlab language.

To identify an optimization oriented model of the Red River Delta, two kind of tools as developed, both in octave/matlab language:

- functions to analyse Mike11 simulation outputs;
- functions to identify reduced models by means of Artificial Neural Networks (ANN), based on a specific library (netlab) from Aston University.

A last type of models required to represent the system: hydropower plants. Thiers representation was obtained implementing a software to optimize within daily and within turbines for each hydropower plants.

In the IMRR project, not only measured timeseries was used as input of the project of alternatives and theirs effects evaluation. The reason is to take into account floods and droughts events, and to analyse the potential and criticality of the Red River system. To define these kind of inflows synthetic timeseries for optimization and simulation operations, was implemented functions to:

- generate synthetic and extreme inflow scenarios;
- downscale climate change scenarios;
- produce the future projection of the inflows using rain-rainfall models.

The first two tools are constituted by functions in octave/matlab language. The third is a group of sub-basin HBV hydrology models.

Beside these headwater inputs, some alternatives use an orienting information to enhance theirs ability to reduce floods event. To perform this result, an estimation of the flood-condition in next days was obtained, via a Fuzzy Empirical Classifier. Also this tool is based on octave/matlab functions.
Mainly, all the designed alternatives are simulated using the Evaluation Oriented Simulator previously described. Using the simulated trajectories, all indicators are evaluated thanks to specific functions and a main program, namely "calc_indicator", developed in octave/matlab language. This software produces either daily values of indicators, yearly and synthetic values for whole the evaluation horizons, representing historical conditions, extreme and projection in the future.

A synthetic way to visualize this kind of results is the Interactive Parallel Plot Suite. It is a javascript tool, based on the D3 library, to explore performance of multiple alternative solutions on a multi-dimensional plot, thus supporting the negotiation process with the institutional stakeholders for the identification of good compromise solutions. To link calc_indicators outputs with this tool, a specific exporting interface was developed.

To compare the optimized policies with normative indication, we developed in octave/matlab also a software to simulate the Reservoir Management Guidelines (both 2011 and 2015 versions).
Bibliography
