D5.3 - Front End Functionalities and Advanced Human-Machine-Interfaces

WP5: Water Cyber-Physical System

30 November 2022

Authors: Dimitris Apostolou (ICCS), Nikos Papageorgiou (ICCS), Dimitra Pournara (ICCS), Gregoris Mentzas (ICCS), Kostas Kalaboukas (MAG), Aziz Mousas (MAG), Efthalia Karkou (TUC), Athanasios Angelis (UOH), Stavros Lounis (AUEB), George Zois (AUEB)

The AquaSPICE project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 958396.

Document Information

GRANT AGREEMENT NUMBER	958396	**ACRONYM**	AquaSPICE
FULL TITLE	Advancing Sustainability of Process Industries through Digital and Circular Water Use Innovations		
START DATE	1st December 2020	**DURATION**	48 months
PROJECT URL	www.AquaSPICE.eu		
DELIVERABLE	D5.3 – Front End Functionalities and Advanced Human-Machine-Interfaces		
WORK PACKAGE	WP5 – Water Cyber Physical System		
DATE OF DELIVERY	**CONTRACTUAL** 11/2022	**ACTUAL**	11/2022
NATURE	Report	**DISSEMINATION LEVEL**	Public
LEAD BENEFICIARY	ICCS		
RESPONSIBLE AUTHOR	Dimitris Apostolou (ICCS)		
CONTRIBUTIONS FROM	Nikos Papageorgiou (ICCS), Dimitra Pournara (ICCS), Gregoris Mentzas (ICCS), Kostas Kalaboukas (MAG), Aziz Mousas (MAG), Chris Dimitropoulos (MAG), Efthalia Karkou (TUC), Athanasios Angelis (UOH), Stavros Lounis (AUEB), George Zois (AUEB)		
ABSTRACT	This deliverable investigates integration between WaterCPS and users by designing a set of front-end and human-machine interfaces. The design takes into account user roles, data requirements and permissions, as specified in WP3. The interfaces provide access to the WaterCPS key functionalities, including the Digital Twin operations, simulation, analytics, LCA and optimisation. Clearly structured, work task-oriented interface design aims to ensure intuitive user experience. Flexibility of interaction techniques (modalities, equipment types and users work situation) are diligently addressed. The deliverable results in a unified set of user interfaces and interaction techniques that can adapt to users at multiple levels and different roles.		

Document History

VERSION	ISSUE DATE	STAGE	DESCRIPTION	CONTRIBUTOR
0.1	25 Oct 2022	Final Draft	Final draft version for internal peer review	Dimitris, Apostolou, Nikos Papageorgiou, Dimitra Pournara, Gregoris Mentzas, Kostas Kalaboukas, Aziz Mousas, Chris Dimitropoulos, Efthalia Karkou, Athanasios Angelis, Stavros Lounis, George Zois
1.0	28 Nov 2022	For submission	Final version following internal review remarks	Klemen Kenda (reviewer)

Disclaimer

Copyright message

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS/ACRONYMS

AI	Artificial Intelligence
API	Application Programming Interface
CS	Case Study
LCA	Life Cycle Assessment
DT	Digital Twin
ICT	Information and Communication Technologies
PSM	Process Simulation and Modelling
RTM	Real Time Monitoring
PSM	Process Simulation and Modelling
WaterCPS	Water Cyber Physical System

1. Executive summary

This deliverable focuses on the interaction between the WaterCPS and users by designing a set of front-end and human-machine interfaces. The interfaces provide access to the WaterCPS key functionalities, including Digital Twin operations, simulation, analytics, LCA and optimisation. Clearly structured, work task-oriented interface design aims to ensure intuitive user experience. Flexibility of interaction techniques (modalities, visualisations and user tasks) are diligently addressed. The deliverable results in a unified set of user interfaces and interaction techniques that support user tasks related to monitoring the water treatment processes.

2. Introduction

Deliverable D5.3 describes the requirements, design and mockups of the WaterCPS advanced user interfaces. WaterCPS integrates the interfaces of all technical components and provides a unified view to virtualization, optimization and water efficiency management services. Note that within the overall AquaSPICE IIoT environment, physical process assets, equipped with sensors, are linked to the RTM platform (reported in D3.5 and D3.6). The RTM platform uses its own interfaces and user interaction modalities (not described herein). The WaterCPS utilises data directly from the RTM platform and provides a unified view to the aforementioned functionalities, which enable its users to simulate, analyse and optimise the water treatment process.

The WaterCPS advanced user interfaces include: (a) Digital Twin interface providing access to the full representation of the knowledge core of the physical system it manages; (b) An interface of the physical water treatment process model; (c) A simulation model of the treatment process; (d) An AI and data analytics interface; (e) An interface to the AquaSPICE optimisation functionalities; (f) An interface to the Life-cycle analysis functionalities. The remaining of the deliverable is structured according to the aforementioned user interface constituents.

WaterCPS integrates a sophisticated context-aware access control, data Integrity and security assurance mechanism, developed in Task 3.4 and reported in Deliverable D3.4 (Data Integrity and Security Assurance in the RTM architecture). The mechanism allows defining fine-rained access control policies that consider user roles and permissions and multiple levels. The reader can refer to D3.4 for a detailed description of issues related to access control, user hierarchy and roles and how the implemented mechanism addresses the users' service and data requirements and permissions.

3. Digital Twin

3.1. Frontend Overview

The front-end UI of WaterCPS has been designed around each pilot's needs. Understanding these needs in order to design these pages was a complex task. The first step was to communicate with the pilot itself, with the aim of pin-pointing the availability and types of data that should be visualized. Further, communication with each individual team responsible for implementing the Analytics, LCA, Optimization and Simulation modules was required in order to define the scope of the interfaces of each component and their interplays. The input and comments received from our partners helped in pinpointing the formats and types of outputs that should be integrated in the frontend interface. At the same time, we took into consideration the end users' needs as expressed in WP1 (D1.2), and kept our UI designs aligned with their requirements.

The overall front-end UI of WaterCPS is designed as a Dashboard (Figure 1). The Dashboard landing page comprises three panes: The status overview, the process diagram and the DT overview. The Dashboard is a highly-customisable front end that displays the information in the form of data, KPIs and graphs pertaining to the water treatment process. The Dashboard is highly customizable according to each user's needs and can be easily configured; it contains a unique configuration of 'cards'. As seen in our platform, these cards include important graphs and KPIs regarding the Digital Twins.

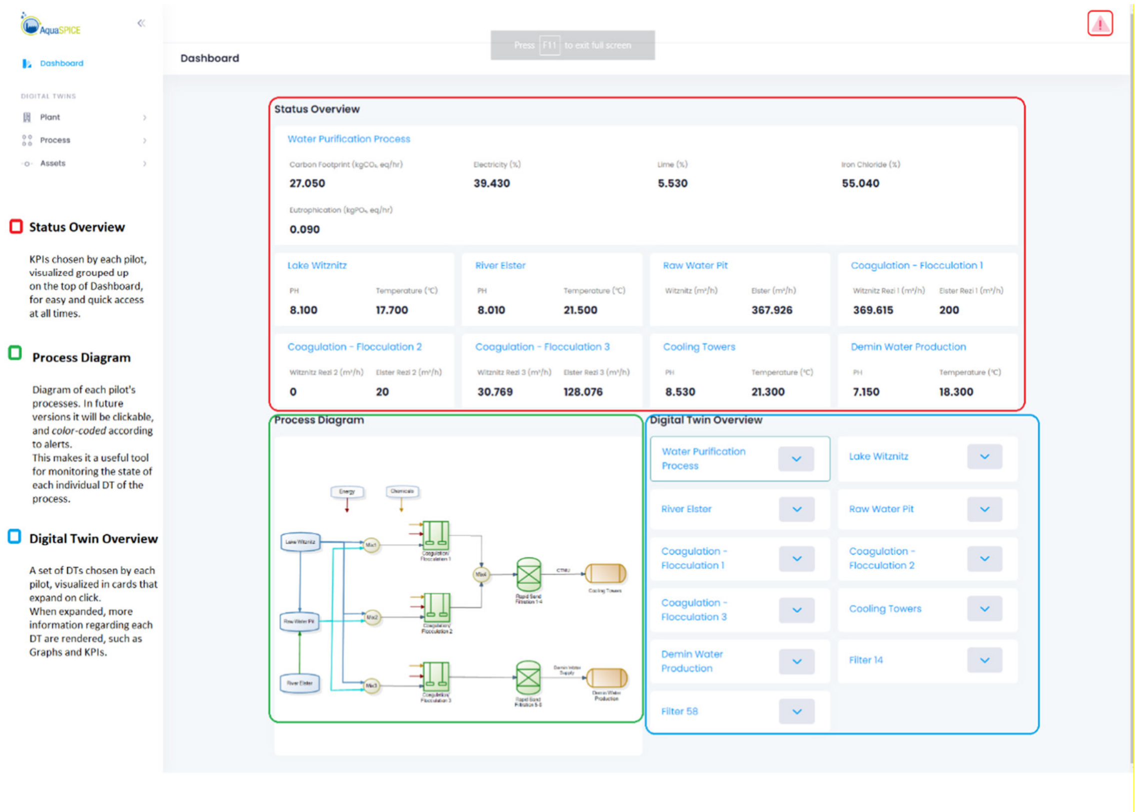


Figure 1. WaterCPS Dashboard

At the top part of the Dashboard (Status Overview), a set of KPI cards is being visualized. This design decision addresses the need of pilots' engineers to monitor quickly and easily specific process parameters. KPIs have been chosen after discussing with each pilot in order to pinpoint their needs. They are deemed important and in need of frequent checking, thus keeping them in a position such as the top of our applications central page was the best option.

Having a process diagram (Process Diagram) representing the water treatment process is useful because not only it helps with connecting the DTs to their actual physical counterparts but also it gives a quick look into the whole process status via color-coding. By making use of the alert system, we recolor each node of the process diagram to signal specific outputs to the end user. For example, if the status of a DT is "OK", the coloring of its node will be green. If an alert is risen for it, with "CRITICAL" status, the coloring will be red. Respectively for "WARNING" status, yellow color has been chosen, etc.

The Digital Twin Overview cards serve as a way for users to create a combination of important DTs, choose as many KPIs to visualize from them, but also choose Graphs and other visualization that would normally take a lot of space. We dealt with the space issue by making these cards expandable, meaning that the user may choose to expand a specific DT card, to view its contents as needed. When they are done viewing information of a DT, users may click on the expansion button again, to close the card and make its visualization smaller, saving space, making it easier to fit more information in the Dashboard. The Dashboard contains links to various pages that are formed by manipulating cards for describing the Monitoring, Analytics, Optimization, Configuration and Simulation interfaces. For example, in the monitor page of the Demin Water Production unit, the cards representing raw data and a graph of PH in the last 4 hours are being rendered (Figure 2).

Figure 2. Demin Water Production Digital Twin Dashboard

Users can navigate from one page to another by clicking on the unit names inside the side-menu (Figure 3).

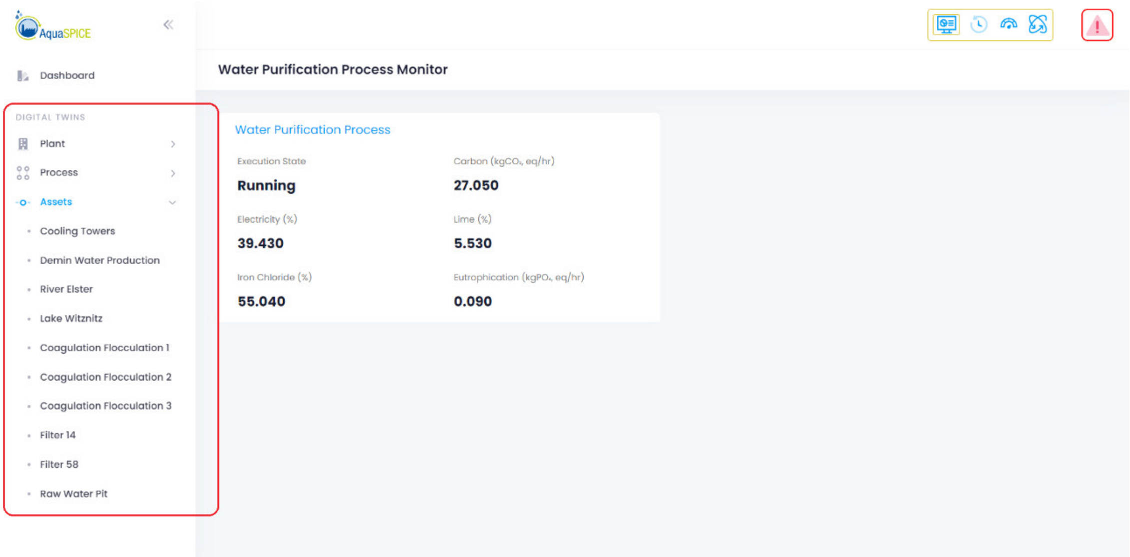


Figure 3. Digital Twin Dashboard side menu

The available functionalities for each unit can be seen in the page-tab menu on the top right of the DT page. After a user clicks on a specific DT unit, the page-tab menu is displayed (Figure 4). More detailed information regarding the outputs and uses of each different page will be explained in subsequent chapters.

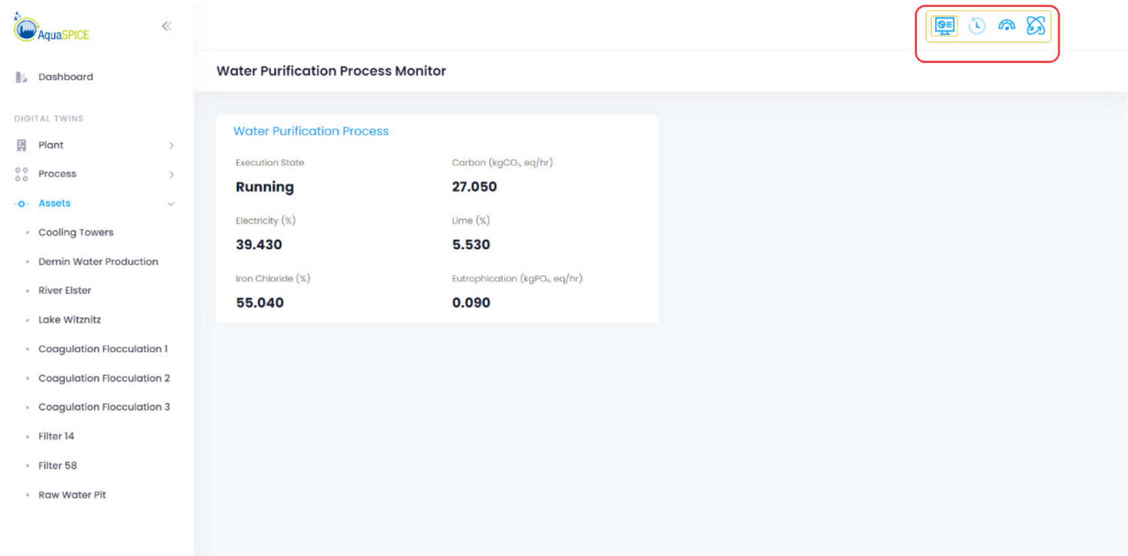


Figure 4. Page-tab menu

The look and feel of the Dashboard is important; while the users navigate through it, they must view consistent visualization that are in harmony with one another and compliment the data appropriately. Having consistency, not only gives a strong identity to the API but also creates patterns for the users to identify important information fast. We've kept in mind all the above while decorating the Dashboard with visualization (html, css etc.). A graph colouring scheme has been selected for the frontend to be consistent with the colour scheme of the project web site and dissemination material. By having the same or close-enough palette of colours, front-end blends in, making the user feel like it is the image of the whole project and not a separate implementation. Moreover, the selected colour palette avoids high contrast elements and strong, bright colours to reduce eye fatigue. Bright, intense coloring has been used only to compliment important information, such as “CRITICAL” alerts. Keeping light coloring through the UI and making use of the strong ones only for important visualizations, is ensuring that the end users will be able to quickly distinguish events that require attention.

3.2. User Interface Configuration

3.2.1. Layout Configurations

Initializing and configuring a new front-end for the AQUASPICE Digital Twin platform, first requires the development team to discuss with involved stakeholders about what kind of visualizations are useful to them, and the way these visualizations should be served in the UI. After pinpointing the specific types of data and the way they need to be presented, the database which stores layout configurations must be configured (Figure 5). The database’s structure is shaped according to the data that will be fed to it.

```
{
    "label": "DT Label",
    "thingId": "DT ID",
    "valueArray": [
      {
        "value": "API/value/path",
        "label": "Value Label",
      },
    ],
    "numberOfGraphs": 1,
    "graphsArray": [
        {
          "tag": [
            "Graph Id Tag"
          ],
          "title": [
            "Graph Title"
          ],
          "threshold": ["2"],
        }
    ],
    "type": "type of card",
},
```

Figure 5. Example layout database serialization

The database is responsible for storing layout configurations which consist of contexts, dictionaries and cards. These configurations allow our front-end platform to render the whole UI correctly, by fetching the data needed from different APIs integrated for the pilot’s case. Behind every page’s layout configuration there is a specific set of structures saved in the database. By making use of the aforementioned database modular structure, we eliminate most of the hard-coding that would be otherwise needed for configuring the user interface according to the needs of a particular application use case.

There is also a second database serving our UI, which is responsible for integrating all external APIs of e.g., the optimisation and analytics components or from bespoke pilot’s case components. This database is built on Eclipse Ditto and serves us by storing and feeding all the sensor readings, KPIs and Digital Twin contexts. At the same time, it accomplishes the role of an orchestrator by handling all the integrations of the calls needed for the front-end to communicate with each external API. For instance, in the analytics case, the graphs and tables are being served as an

HTML static file output. The front-end UI then calls the Ditto platform, in order to fetch these html outputs. The same procedure is followed for fetching data from Optimization, Simulation and LCA services.

3.2.2. UI Components

3.2.2.1 Cards structure

The card-structure configuration describes the types of visualizations that will be included in each card, and the output values assigned to it. All the value identifiers and graph-tag identifiers along with their according labels can be found in the card-structure (Figure 6).

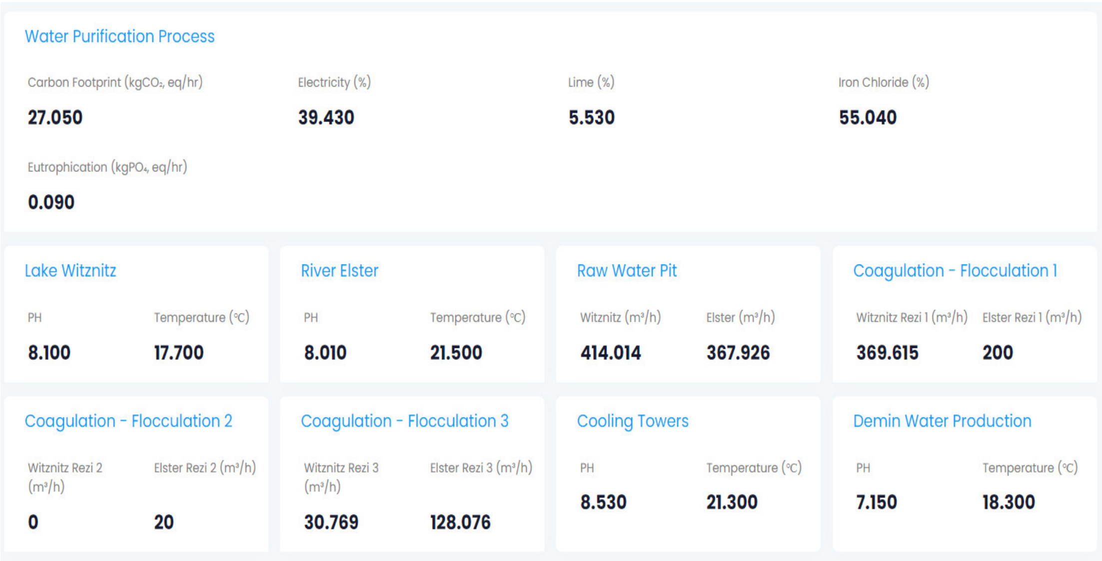


Figure 6. Card structure

3.2.2.2 Graphs structure

Another type of structure is the Graphs structure. This structure holds important information required for the front-end to render different types of data in the form of linear graphs, bar graphs etc. (Figure 7).

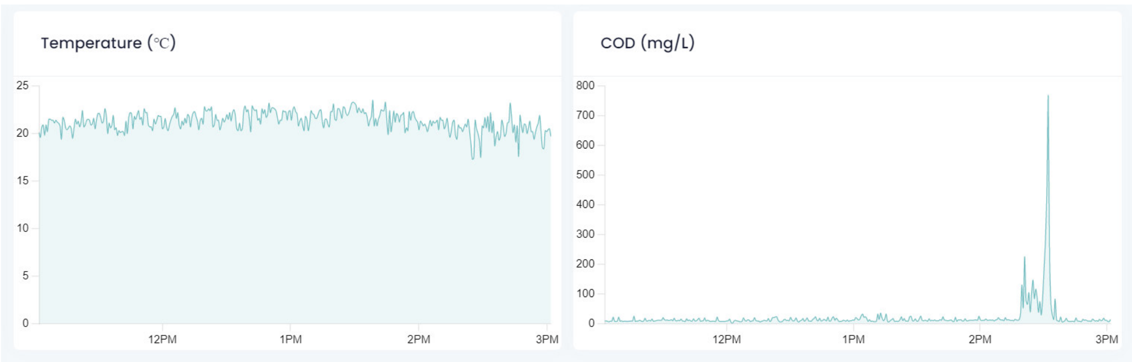


Figure 7. Graph structure

A variety of graph types is supported, including gauge graphs showing percentages of completion (e.g., capacity of a container (Figure 8). The library used for producing these graphs is Chart.js.

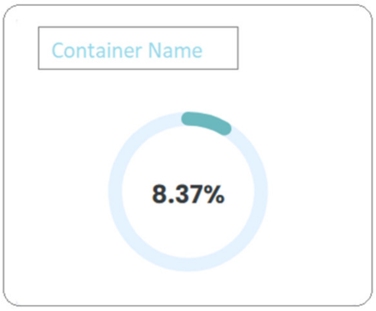


Figure 8. Example of gauge graph

3.2.2.3 HTML imports

Another way for visualizing data is by directly importing and rendering HTML outputs. Third-party tools and applications are capable of producing sophisticated visual results that are not always available in chart.js. These tools may be bespoke for a particular application use case and typically make their visualisations available as HTML pages. Thus, having the ability to directly import graphs from HTML pages and render them in the frontend not only provides additional visualisation capabilities, but also reduces the integration effort and saves integration time. In order to do that, two things are mandatory.

- The API which the results come from needs to export HTML instances.
- The front-end UI needs to import these instances by using HTML methods such as iframe. Additionally, configurations need to be properly implemented in order for the HTML imports to look harmonized with the pre-existing components of each page.

An example of such imports can be seen in the Analytics page components, which are directly rendered through the AI and Analytics component described in section 4.

3.2.2.4 Process Diagram

Based on the pilots' inputs, the rendering of process diagrams depicting the water treatment process was required for the front-end. This task was completed by the BPMN.io library supporting the design and rendering of process graphs (Figure 9).

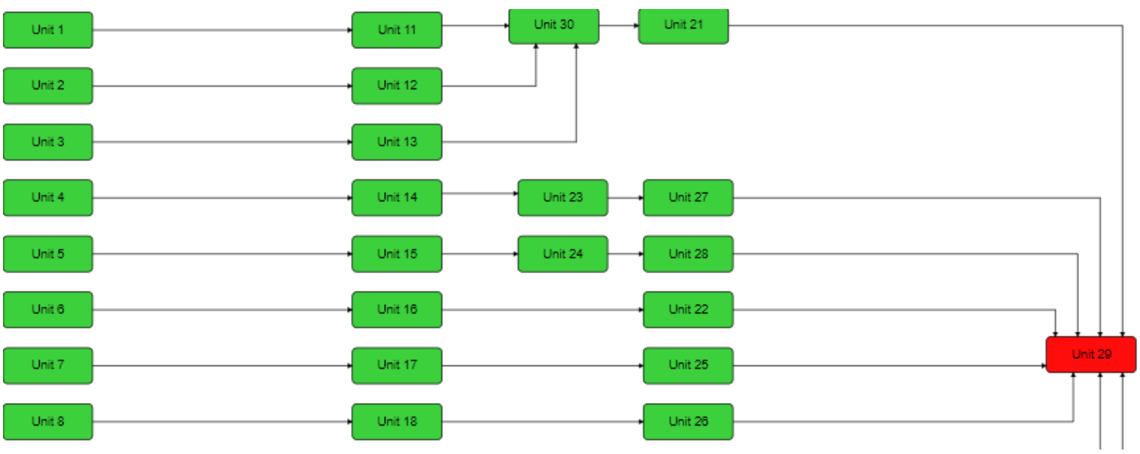


Figure 9. Water Treatment Process graph

4. AI and Analytics UI

This section describes the AI and Analytics user interfaces. The functionalities described are based on preliminarily, basic descriptive and predictive analytics methods. These methods will be revised and significantly extended in the scope of the ongoing Task 4.5 'AI Inference for Water Efficiency Problem Detection, Route-Cause Analysis and Remedial Action'. The final set of functionalities will include more advanced AI and analytics methods, e.g., for anomaly detection, nowcasting, classification models (if needed).

Section 4.1 describes the 'run-time' interfaces that enable users to view the results of the AI and Analytics component of the WaterCPS. Section 4.2 describes the Model Development and Experimentation workbench, which is the 'design-time' environment for developing AI and Analytics models.

4.1. AI and Analytics User Interfaces

The WaterCPS user can get an overview of the AI and Analytics results by clicking on the Analytics button, where available in the Assets pages of the WaterCPS (Figure 10). Depending on the implemented AI and Analytics models implemented for the specific asset, the button will display descriptive, predictive or both types of analytics.

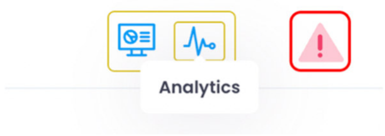
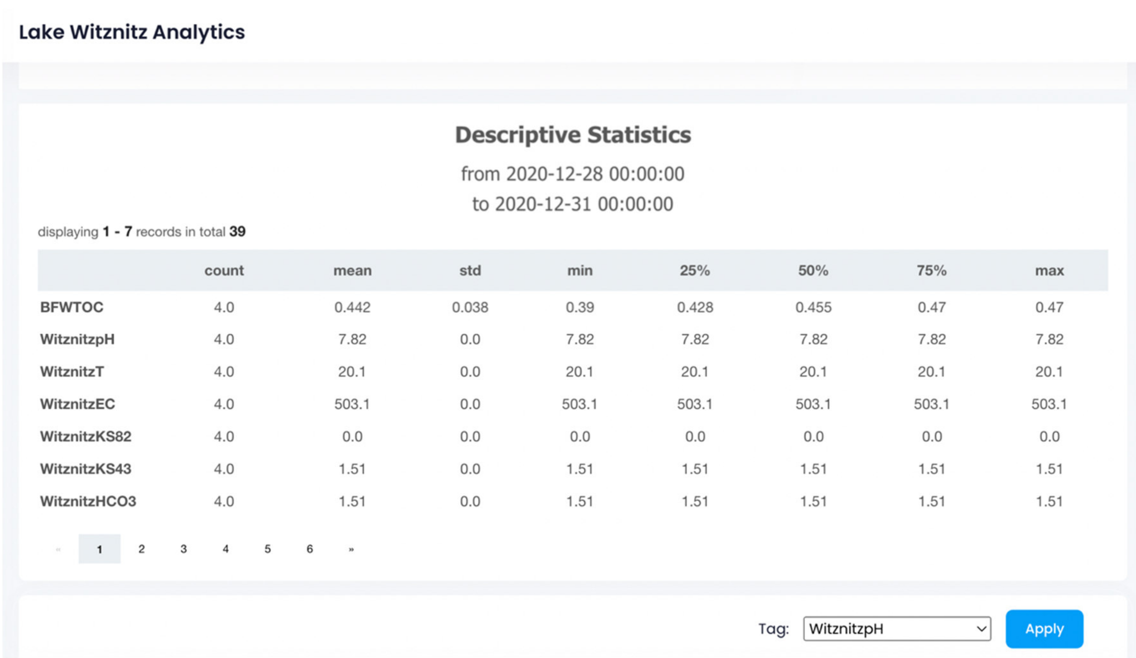


Figure 10. Button for displaying AI and Analytics results

Descriptive analytics show key statistics such as average, minimum and maximum, quartiles, and standard deviation for selected variables and time windows. Results are shown either in tabular form (Figure 11) or as violin charts (Figure 12).


Lake Witznitz Analytics

Descriptive Statistics

from 2020-12-28 00:00:00
to 2020-12-31 00:00:00

displaying **1 - 7** records in total **39**

	count	mean	std	min	25%	50%	75%	max
BFWTOC	4.0	0.442	0.038	0.39	0.428	0.455	0.47	0.47
WitznitzpH	4.0	7.82	0.0	7.82	7.82	7.82	7.82	7.82
WitznitzT	4.0	20.1	0.0	20.1	20.1	20.1	20.1	20.1
WitznitzEC	4.0	503.1	0.0	503.1	503.1	503.1	503.1	503.1
WitznitzKS82	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
WitznitzKS43	4.0	1.51	0.0	1.51	1.51	1.51	1.51	1.51
WitznitzHCO3	4.0	1.51	0.0	1.51	1.51	1.51	1.51	1.51

« 1 2 3 4 5 6 »

Tag: WitznitzpH Apply

Figure 11. Descriptive Analytics – Tabular View

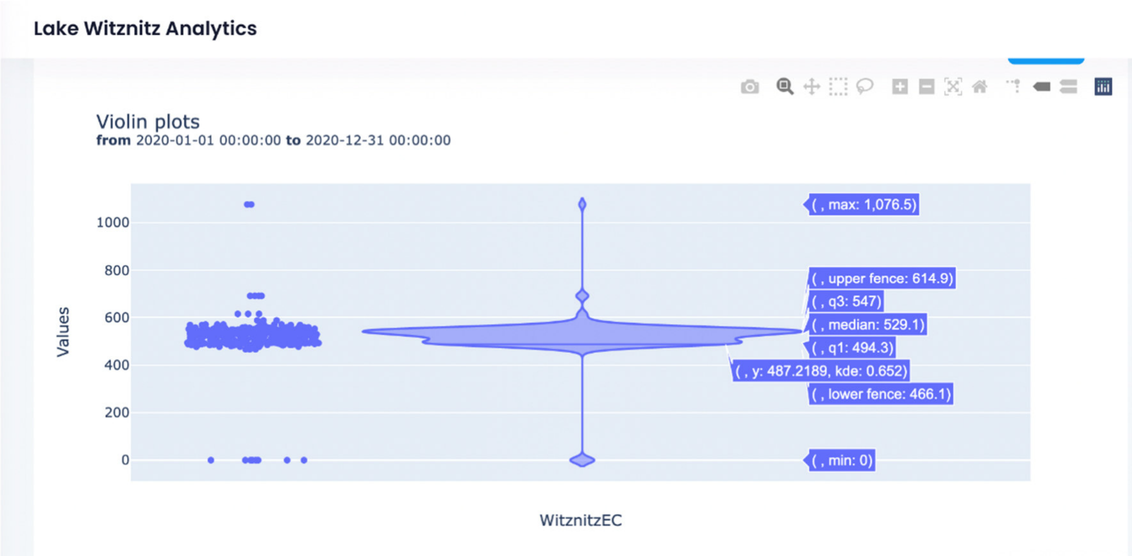


Figure 12. Descriptive Analytics – Graph View (Violin Chart)

Predictive analytics show in the same view actual values and predicted by the AI model values of selected variables. Moreover, predictive analytics display the predicted values for the next two days (Figure 13).

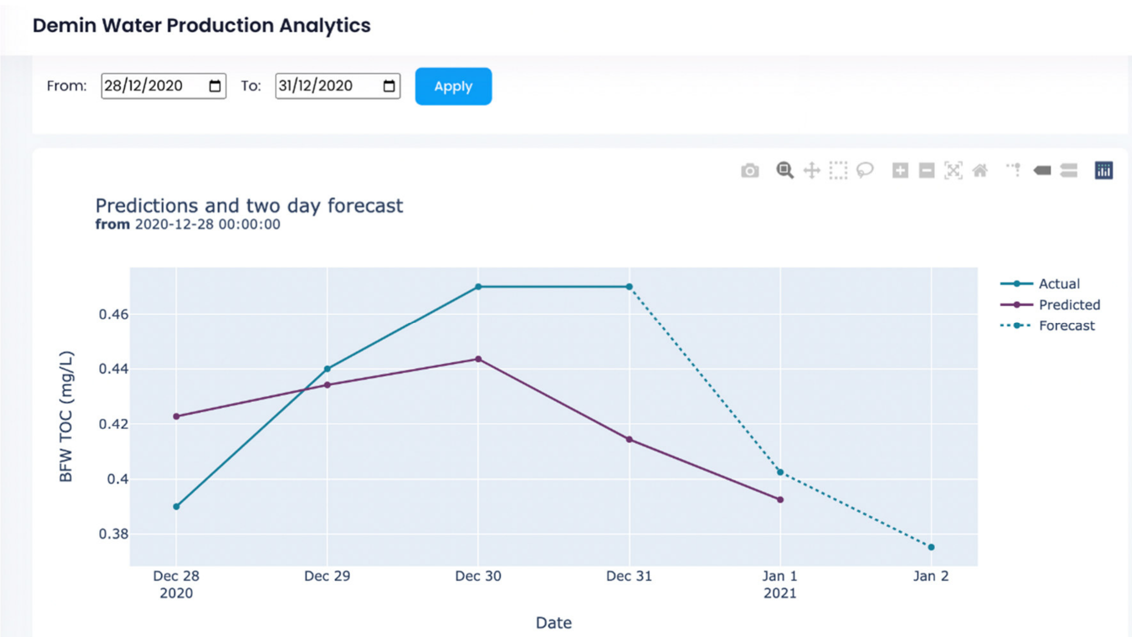


Figure 13. Predictive Analytics

4.2. AI and Analytics Models Development and Experimentation

An AI and Analytics model development process involves model building, training, hyperparameter tuning and evaluation aiming to find the optimal model. Being a deeply experimental process, it requires multiple trials and comparisons between different models. The complexity of the process poses the need for comprehensive system management services, including AI configuration functions, diagnostics and workflowing capabilities for smart AI operations. To address the need, the workbench interface described in this section allows analysts and data professionals to perform complex analysis and experiment with Machine Learning (ML) models; its intuitive interface is also suitable for non-technical users. Through a Graphical User Interface, the user can perform various data analytics tasks under the following major categories:

- Experiment execution
- Experiment tracking
- Experiment comparison and analysis
- Hyperparameter tuning for ML models
- Parameterization of ML models
- Best model selection for productization

The examples described below depict the Boiler Feed Water Total Organic Carbon prediction for the next two days model. The data analytics pipeline consists of two operations: i. Preprocess, and ii. Train. These operations will be described in detail in D4.5 "AI Inference Tool for Water Efficiency Problem Detection and Remedial Action". In a nutshell, during the preprocess step, the data is cleaned from static values and rows with missing data. Remaining missing values are treated using linear interpolation method. The train step includes feature selection, data fusion (finding additional contextual data that can improve the models as well as training model selection (e.g. linear regression, random forest, gradient boosting, LSTM, etc.) as well as actual model training.

The user can execute stand-alone operations or pre-defined data analytics pipelines. The experiment's attributes are recorded throughout its lifecycle and are presented in the user dashboard. The attributes include useful information such as the operation's description, the status of the experiment (completed, fail, running), timestamps, flags and other artefacts that were set for the specific experiment.

The dashboard initializes to the most recent operation. However, the user can select an operation from the list to view its metadata. In the "overview" tab the selected run's meta-information are presented; such information includes the run's ID, operation name, status, timestamps, flags (hyperparameters and parameters), scalars (metrics) and environment variables (Figure 14). In case files were utilized, their details will be presented in the "files" tab (Figure 15). Outputs of the code execution (if any) are displayed in the "log" tab (Figure 16). In the "compare runs" screen, a table with brief run summary is presented (Figure 17).

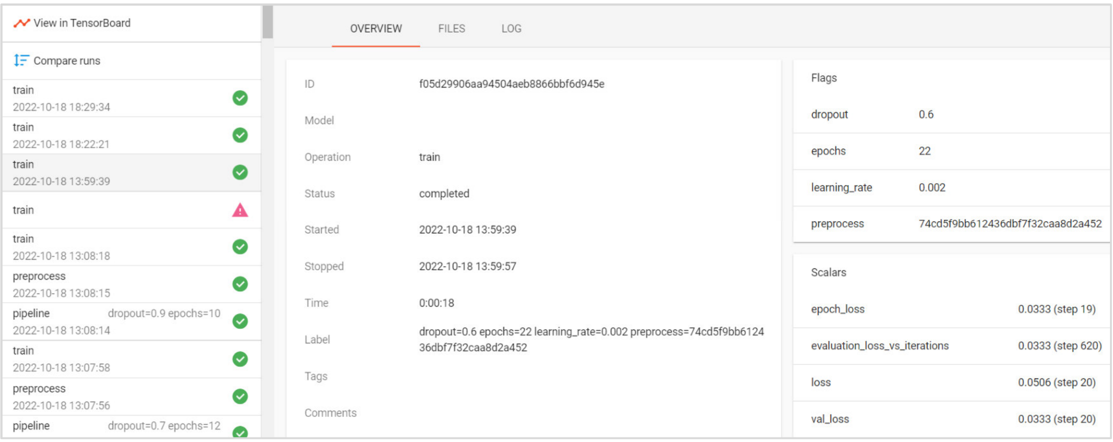


Figure 14. Experiment Overview

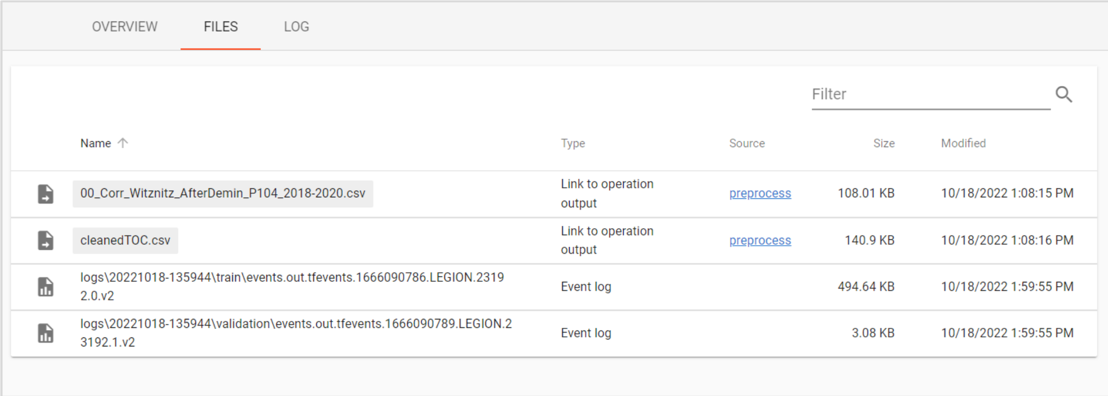


Figure 15. Experiment Files

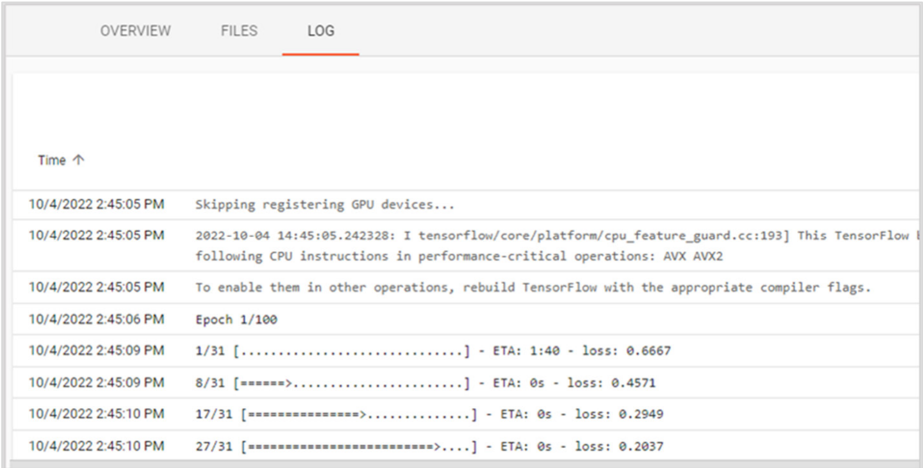


Figure 16. Experiment Log

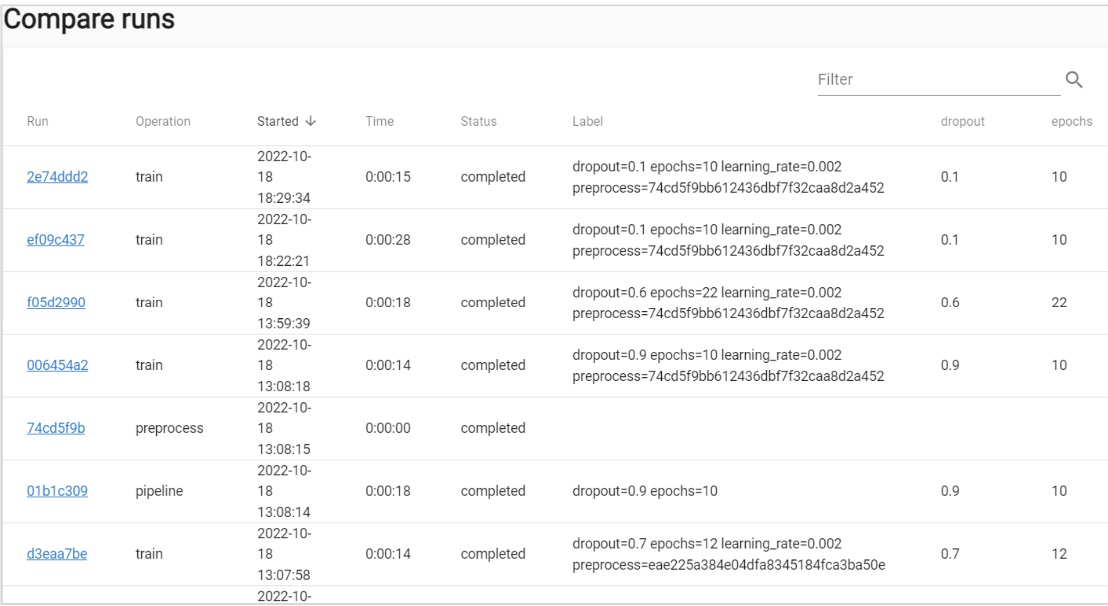


Figure 17. Compare Runs

Advanced users can apply the following techniques to find the optimal hyperparameters configuration for the ML models:

- Manual search: User specifies the candidate hyperparameters sets and the number of trials
- Grid Search: An exhaustive search method. The user specifies a range of values per hyperparameter and then one model is constructed and evaluated for each combination of parameters.
- Random search: Includes the generation and evaluation of random inputs for the hyperparameters over a uniform or a log-uniform distribution. The range of values is specified by the user.
- Bayesian optimization: This approach builds a probabilistic model (surrogate) of the objective function that then suggests the most promising hyperparameters to be evaluated in the true objective function. The supported surrogate models are gaussian processes, random forests, gradient boosted regression trees and random optimizers.

The hyperparameters that can be tuned in an LSTM model include: number of neurons; number of hidden layers; number of epochs; batch size; activation function; learning rate; weight initialization; decay rate. In the example below, the dropout and the number of epochs is optimized using Bayesian optimization with Gaussian processes.

Apart from the compare run option, the user can view the experiment's metadata inside the dashboard from the relevant menu option. Interactive diagrams and insights help users with experiment's comparison, hyperparameter optimization and deeper understanding of the models. Indicative diagrams include various views of the hyperparameters (distributions, histograms, scalars, time series, and graphs). Filtering capabilities are provided across the view and are applied to the active tab. For example, the table view lists the runs, their hyperparameters, and their metrics (Figure 18). In the Parallel Coordinates View (interactive), every run is represented by a line going through an axis for each hyperparameter and metric. This view is suitable to identify the most important sets of hyperparameters (Figure 20). The Scatter Plot Matrix View (interactive) visualizes the comparison between the hyperparameters and the metrics. This view is useful to identify correlations between the parameters (Figure 19). A suitable set of candidate hyperparameters for the example scenario could be dropout = 0.6 for 22 epochs.

TABLE VIEW | PARALLEL COORDINATES VIEW | SCATTER PLOT MATRIX VIEW

Trial ID	Show Metrics	dropout	epochs	learning_rate	loss	time	val_loss
006454a2 train	☐	0.90000	10.000	0.0020000	0.21620	14.034	0.10550
09435050 train	☐	0.50000	20.000	0.0020000	0.051200	14.687	0.034500
1c2dc59c train	☐	0.90000	9.0000	0.0020000	0.24700	13.074	0.11030
2799538e train	☐	0.80000	23.000	0.0020000	0.066000	17.515	0.035800
2bfe8f56 train	☐	0.40000	10.000	0.0020000	0.051000	13.563	0.036200
2e74ddd2 train	☐	0.10000	10.000	0.0020000	0.024300	15.798	0.038200
30736234 train	☐	0.70000	29.000	0.0020000	0.10370	13.618	0.042100
3c0b1345 train	☐	0.60000	9.0000	0.0020000	0.084700	13.306	0.038600
4f20acd8 train	☐	0.50000	19.000	0.0020000	0.051200	14.871	0.034500
6141ee35 train	☐	0.60000	13.000	0.0020000	0.065900	13.903	0.037500
782c43e0 train	☐	0.60000	22.000	0.0020000	0.050600	15.410	0.033300
9cfdf9e4 train	☐	0.40000	29.000	0.0020000	0.033100	15.577	0.033900

Figure 18. Hyperparameters Table View

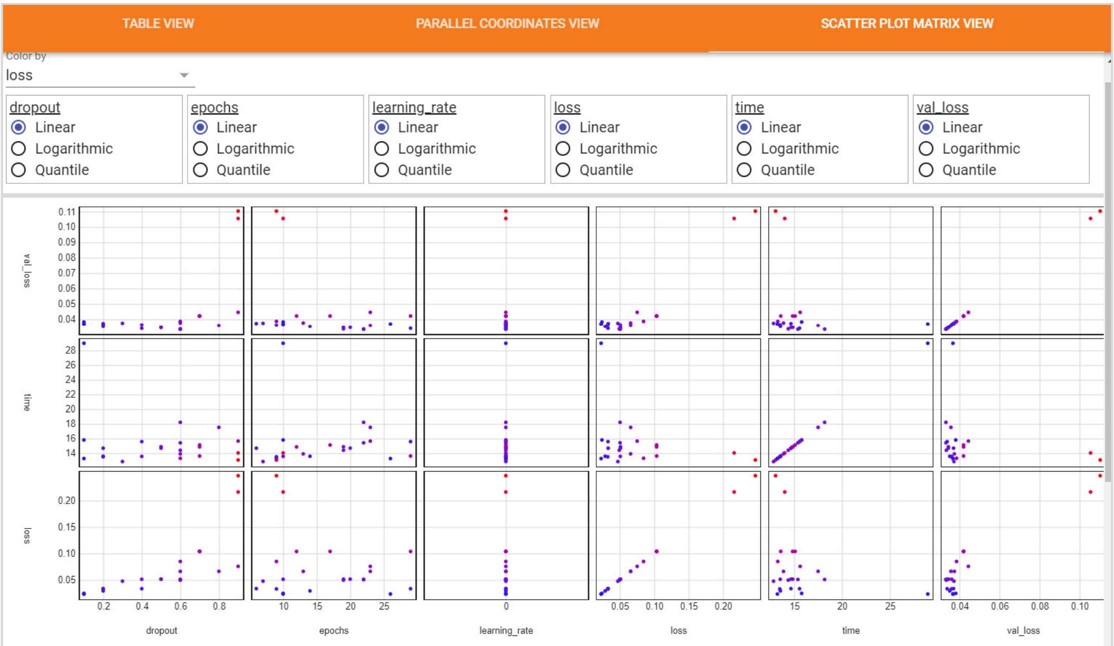


Figure 19. Hyperparameters Scatter Plot Matrix View

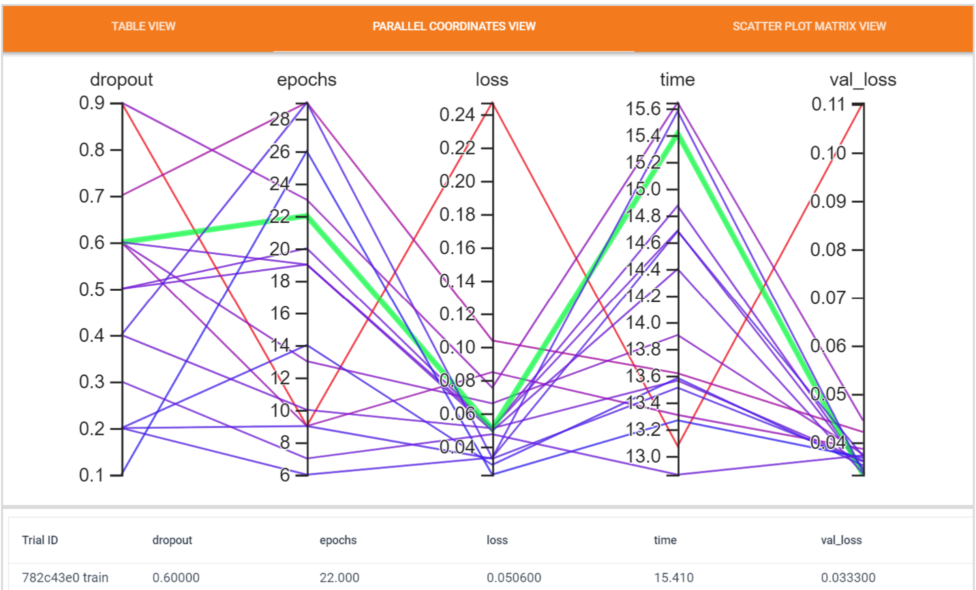


Figure 20. Hyperparameters Parallel Coordinates View

5. Simulation and Modelling UI

5.1. User Interface Configuration

5.1.1. Layout Configurations

The selection and utilization of the appropriate tool to model and simulate all the components entails the determination of the specific requirements of each case study (CS). Firstly, the development team has to clarify with the Case Study Leader (CSL) the existing water treatment technologies in the industry (As-Is Scenario) as well as the processes that are going to be implemented in the future (To-Be Scenario). In this regard, water flows (input, output, reuse and recycle) are pointed out. After that, it is necessary to determine the industrial pollutants of interest in each water treatment technology. Furthermore, the layout configuration incorporates the consumption of chemicals, energy, and the production of by-products, such as sludge and brine.

The Process Simulation and Modelling (PSM) Suite, which is developing in the context of Task 4.3, consists of two parts; a stand-alone tool (PSM Tool) that is used to develop the model representation and simulate an industrial process along with all the components and their interrelations and an API (PSM API) that accelerates the integration of the tool with other services, such as Optimization, Analytics and the AquaSPICE Dashboard, eventually forming the Water Cyber-Physical System (WaterCPS) ecosystem. The PSM Tool presents the entire transformation process of the raw materials into valuable products, which engage the attention of the end-users. This is achieved through the modelling of the flow of materials and energy, the consumed quantity of chemicals and the flow of products in one or a series of production processes. The transformation process is based on the process modelling, which is made using specification factors, scripts or AI models. Overall, the PSM tool is utilized to present the process flow diagram of an industry with a graphical representation, design the process models, specify the flows of materials and energy, interrelate the input with the output, calculate all the involved material flows and present the final results in various forms (discrete values, in tabular format, etc.). Figure 21 presents the generic interface of the PSM Tool.

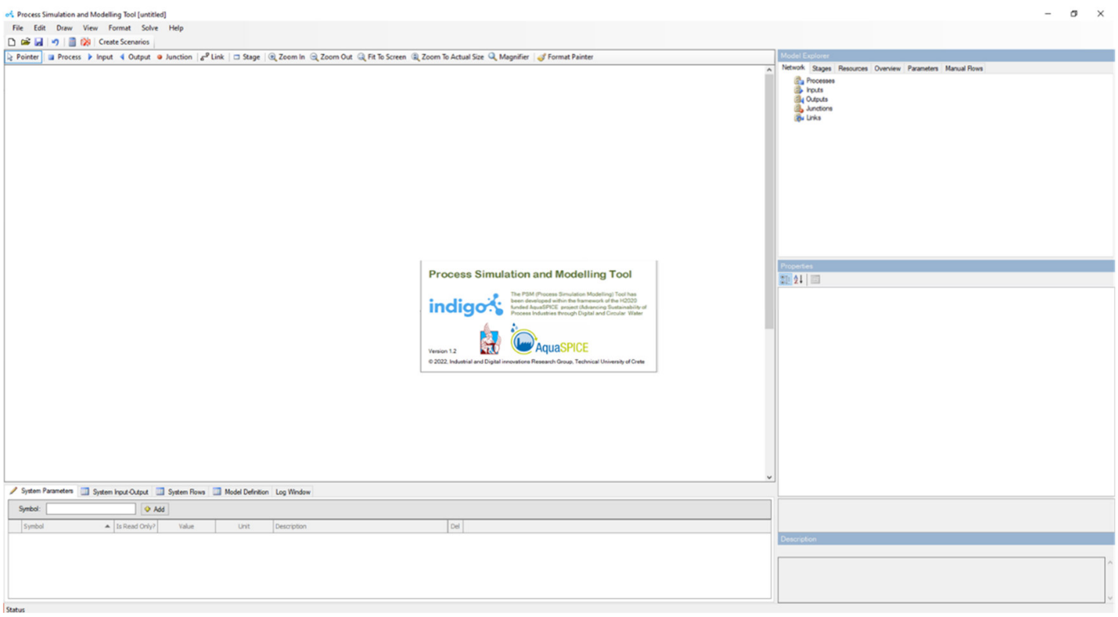


Figure 21. PSM Tool Interface

5.1.2. Components

The PSM Tool uses two types of vertices, processes and places. On the one hand, processes are fed with materials, the so-called inputs, which are transformed into new materials or end-products, the outputs. On the other hand, places are utilized as storage for various resources. They are classified into input nodes, output nodes, and junctions, representing the sources that flow towards a process, sources that exit from a process, and places for the connection of two or more processes, respectively. All the elements of the PSM Tool are connected via “Links”. The system consists of flows that refer to:

- Water-related materials, such as freshwater sources, wastewater, etc.
- Resources that are used in the processing units, such as chemicals, other raw materials, energy, etc.
- Products and by-products produced by the processes, such as sludge, reclaimed water, brine, etc.

Taking as an example the case study of the Dow Chemical plant in Boehlen, Germany, the main dashboard of the WaterCPS platform is shown in Figure 22. The process flow diagram is located at the bottom left side of the dashboard and includes all the water treatment technologies, types and quantity of consumed chemicals, the required energy, and the generation of products (water, sludge).

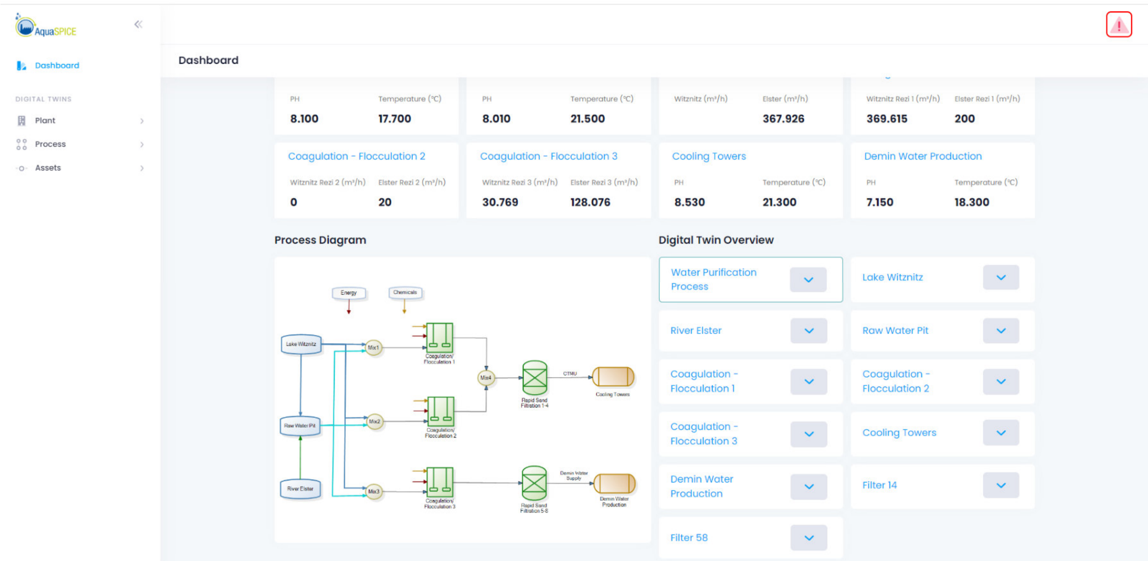


Figure 22. Dashboard of modelling and simulation in the WaterCPS platform

The entities used to develop the user interface for the modelling and simulation of the Dow Boehlen CS constitutes of various vessels, mixing, fluid contacting vessels, tanks, membranes, and columns. In addition, input nodes, junctions, and output nodes are used.

5.1.2.1 Process Flow diagram

Water and wastewater treatment aims at the production of a final product with specific quality and quantity requirements. Industrial processes modify the composition of the input materials by transforming them into outputs with different quality characteristics. Material Flow Networks (MFNs) are utilized to model and simulate the whole system. The consumed and produced quantities of resources are calculated along with the quality.

Regarding the Dow Chemical plant in Boehlen, input materials are fresh water, wastewater, contaminants, chemicals, and energy. The quantities of fresh water, wastewater and chemicals are measured in m^3/h or kg/h. Energy is described in kW. The quality parameters comprise the industrial contaminants and are measured in mg/L, mmol/L, μS/cm or NTU, according to the type of the pollutant. The water and wastewater treatment technologies, both existing and to-be implemented, include, among others, coagulation/flocculation, rapid sand filtration, and reverse osmosis. As end-products are considered the water (as make-up water for the cooling tower and the production of demin water), sludge, and brine (removed from the reverse osmosis).

The PSM Suite, which is currently developing in the context of Task 4.3, has already been implemented for the case study of Dow Boehlen. The integration of this tool in the WaterCPS platform is feasible because of the tool's functionalities. However, it is still ongoing and will be completed soon. In this regard, the following figures present the interface in the PSM for both scenarios (As-Is scenario, To-Be scenario).

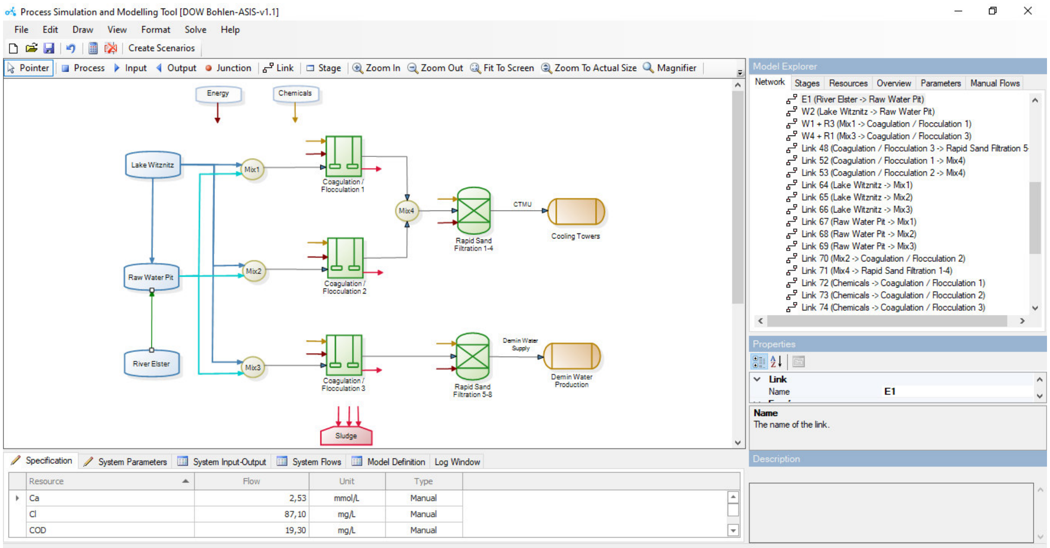


Figure 23. PSM Tool interface for the As-Is scenario in CS#1B

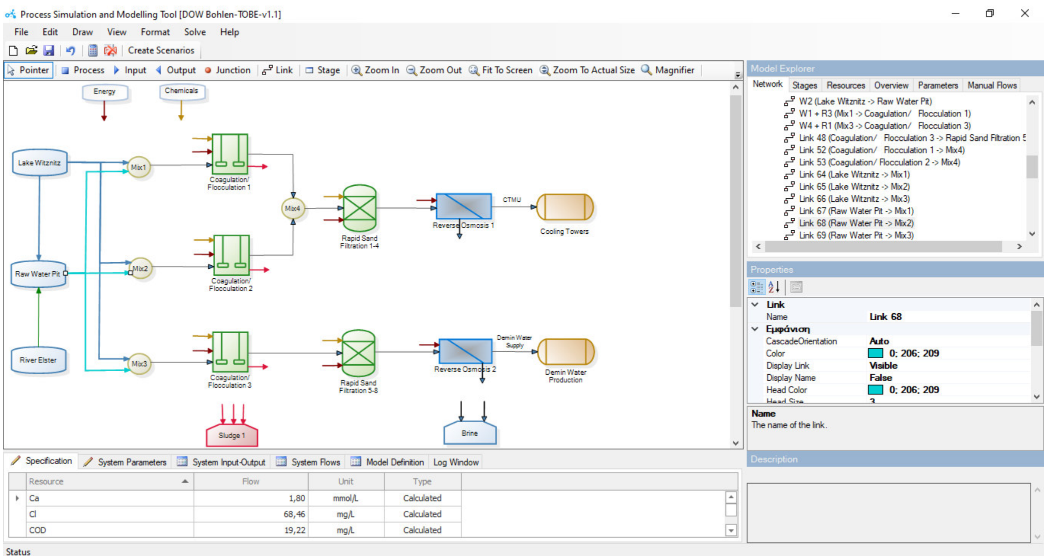


Figure 24. PSM Tool interface for an option of the To-Be scenarios in CS#1B

5.1.2.2 Simulation results

After the integration of the PSM Tool into the WaterCPS platform, the user will have the possibility to simulate the whole production chain. This way, the simulated results will be presented in the corresponding section of the WaterCPS platform, as shown in Figure 25. The simulation of the process in question will be possible through the Simulation Tab of the WaterCPS dashboard and the user will not have to interact with the standalone PSM Tool for simulation purposes; however, there is still the need to use the standalone tool for the process model creation. Unknown values of quality and quantity parameters will be calculated based on iterative algorithms.

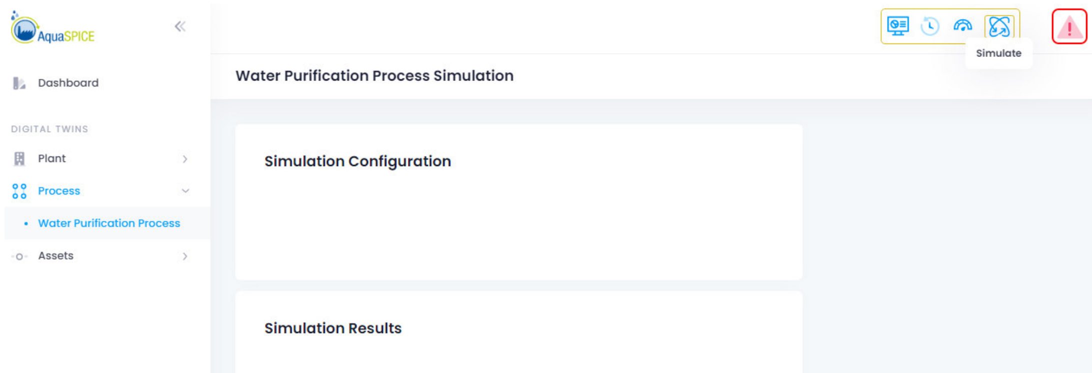


Figure 25. User interface for the simulation results in the WaterCPS platform

An example is presented, as it is in the PSM Suite, which is developing as part of Task 4.3, to make the simulation results more comprehensible. This example refers to reverse osmosis process, with input flows towards the reverse osmosis membrane being known (calculated by PSM). By performing a simulation, PSM Tool is able to calculate the output flows that leave the reverse osmosis process. In this regard, the flow towards the cooling towers and the brine production are calculated, as presented in the red outline of Figure 26. In the left column are the input flows and in the right column are the output flows.

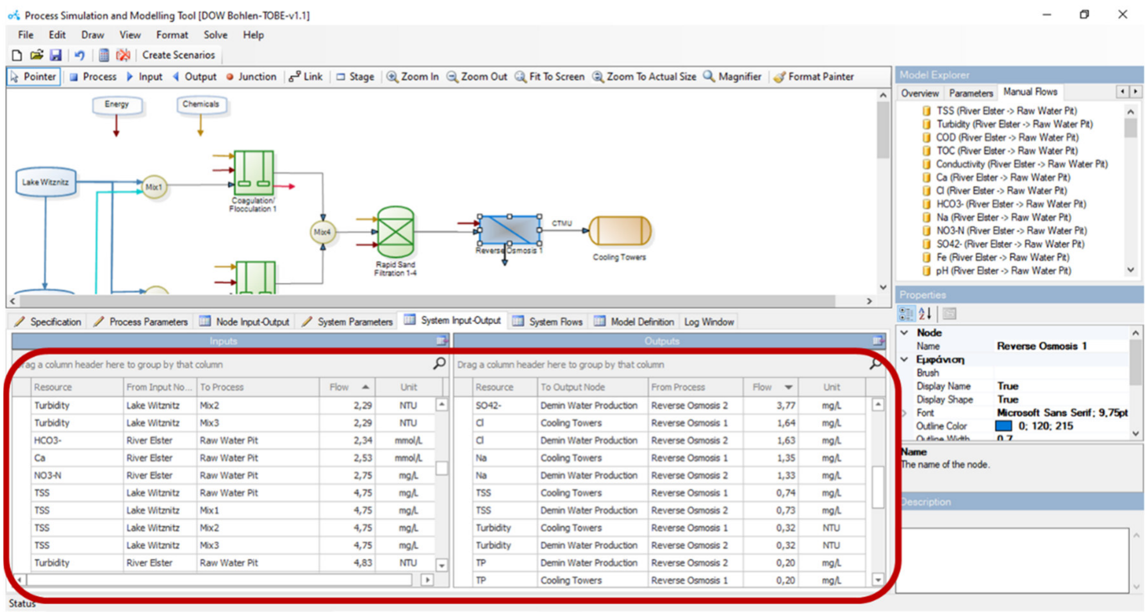


Figure 26. Simulation results as shown in the PSM Tool regarding reverse osmosis membrane

6. Optimisation UI

The optimization module will provide the decision makers of the different AquaSPICE pilots with the support needed in order to respond to their optimization needs at that point in time, given the state of their system and on the issue that they may want to resolve. Thus, the optimization module will be built to accommodate the specificities of each different case that requires optimization within AquaSPICE. This is in accordance with the typical approach for optimization, where in order to achieve high quality solutions in a time-efficient manner, both the mathematical model and the solution method may need to be tailor-made to meet the needs of the specific problem.

To this end, with respect to the implementation of the Optimization module in AquaSPICE, “Pilot Templates” will be provided for each pilot that requires optimization. Each “Pilot Template” will prescribe all the data needed by the optimization module for that particular instantiation (pilot). The Optimization module will thus be able to receive requests from the different pilots, through the WaterCPS in an adhoc and pilot-tailored manner and respond accordingly. Hence, upon completion of the Optimization process, the output (i.e., the optimized solution to the provided problem) will be returned to the WaterCPS in order to be further consumed. For example, the output may be presented to a decision-maker through the respective UI part of the overall front-end system or be consumed by a simulation module to run further analysis and experiments.

The basic functionalities provided by the optimization module are offered to the rest of the AquaSPICE WaterCPS modules through a set of REST services supporting interaction relevant to:

- *ConfigureModel*: ConfigureModel is a back-end operation that enables the WaterCPS to provide the Optimization module with the schema of the water purification process and (if required by the pilot) the corresponding production process.
- *OptimizeInstance*: This is the main operation of the Optimization module. It requires the WaterCPS to send an optimization request along with the corresponding data for the instantiation of the problem as well as related optimization settings and/or objectives. Once a solution is obtained, it is returned to the WaterCPS including both the suggested course of action with respect to the examined problem as well as some basic KPIs related to the proposed solution.

ConfigureModel is utilized at design-time. During the set-up of the WaterCPS and its corresponding modules for a given instantiation (pilot), the Optimization module will be provided with the process schema in a predefined format; the modelling format of the PSM tool provided by TUC (and exported in JSON) will be utilized for this purpose. This format enables modelling (via node and link objects and their attributes) the production or purification processes, pipes, valves, pumps, tanks and clean water sources that comprise each pilot. By receiving the corresponding schema, the Optimization module will be set to handle that particular process and solve the corresponding optimization problem, once provided with such a request and the corresponding instantiation data. Note that ConfigureModel is a back-end operation and hence will not include a User Interface.

As mentioned above, with respect to OptimizeInstance, the request will include:

- Input instantiation data: This may include e.g., initial water flows, open/closed pumps, the concentration of impurities in a given tank, the purification capabilities of a given process etc.

- Optimization settings and objectives: This may include different objectives that may be targeted by optimization as well as settings related to the time horizon provided to optimization for optimizing the corresponding water management and purification process or even the time-frame given to optimization to obtain an optimal solution.

The data requirements for OptimizeInstance will be prescribed in detail in the corresponding Pilot Template for each case treated within AquaSPICE. Further, OptimizeInstance implies four different User Interfaces that are required to use it. More specifically:

- Optimization Request Configuration UI: Through this interface (see Figure 27, left), the optimization toolkit interacts with the user for the instantiation of optimization request as well as the reception of data. Hence, the user may be able to select among different objectives to optimize (depending on the case), provide the time horizon to be examined, specify particular issues related to the problem at hand, etc. Of course, after providing all required settings, the user will be able to request the Optimization module to run, in order to solve the provided problem. The WaterCPS will then provide the OptimizeInstance with all data provided by the User as well as all other data that it will need to retrieve either from its DB, other data sources or other WaterCPS modules (as prescribed in the corresponding Pilot Template).
- Optimization Response UI: This interface (see Figure 29) is responsible for presenting the Optimization output (i.e., the response of OptimizeInstance) to the User. This will include a detailed presentation of the suggested course of action with respect to the examined problem (e.g., use this process for this much time, open this valve and provide the system with clean water for this much time, etc.). It will also provide some basic KPIs related to the proposed solution, e.g., the minimization of contamination factor for a single or multiple pollutant, the total fresh water intake from the water sources, the minimization of energy consumption over treatment processes.
- Optimization History UI: This interface (see Figure 27, right) provides a list of all previously executed optimization calls. The user can hence browse through previous instances and select a single instance to examine.
- Optimization Historic Instance UI: This interface is activated once the user selects a specific instance to view from the Optimization History UI list (see Figure 27, right). This UI includes both the Optimization Request Configuration data (see Figure 28) that were given to the OptimizeInstance service as well as the Optimization Response data that the OptimizeInstance service returned. Thus, the user is able to examine what was provided to the Optimization module and what course of action was suggested by the Optimization module respectively.

Relevant to the Optimization Request Configuration UI and additionally to the prescribed "Templates" (which are directly taken from the WaterCPS) the user will be able to provide the remaining information needed for an optimization request to take place. Indicatively, for the Dow Bohlen case the different configurations presented in the following figure will be, Configuration 1: Process Model (with detailed configuration of treatment processes' parameters per pollutant of Interest), Configuration 2: Optimization Objective (e.g. minimize fresh water intake from River Elster) and Configuration 3: Time Horizon, (e.g. 1 Day). Upon providing the above information the end user proceeds to initiate an Optimization cycle.

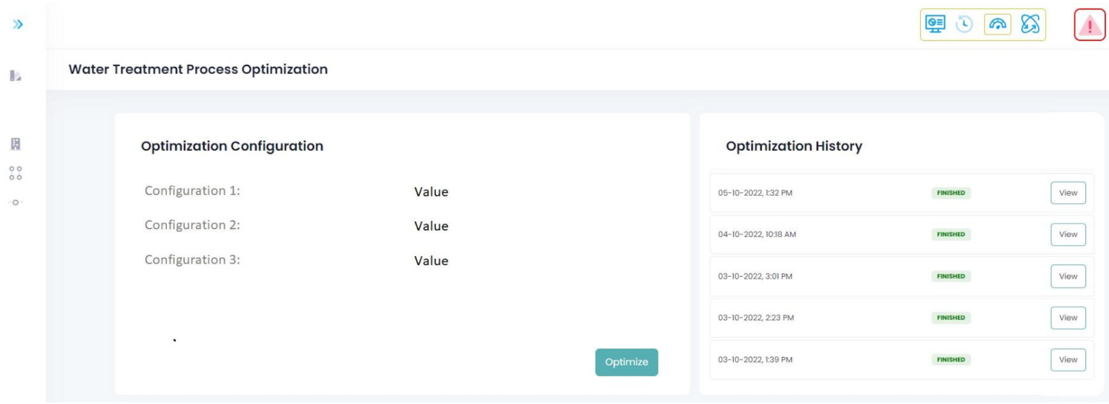
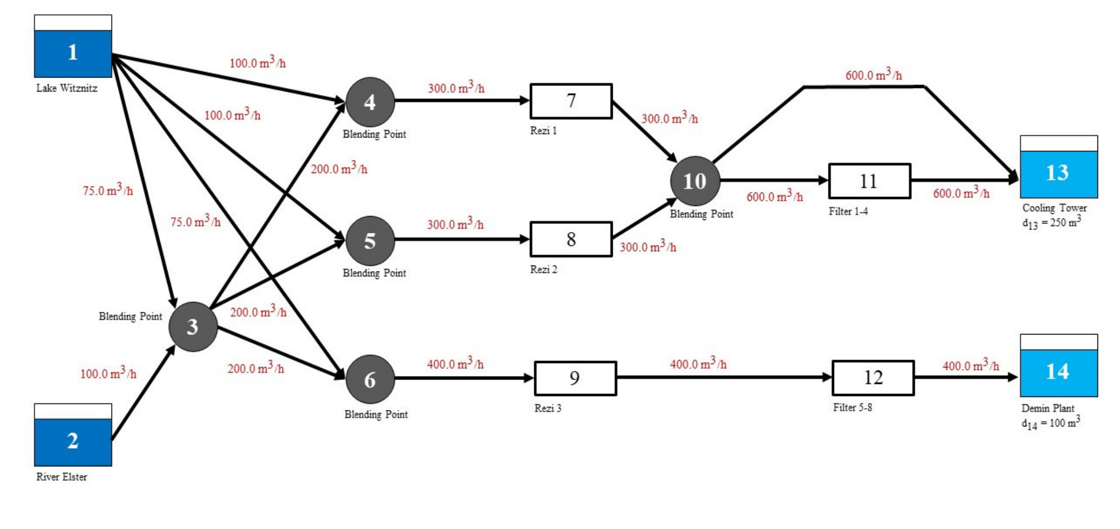


Figure 27 Optimization Configuration and Optimization History

Based on the internal composition of the Optimization module and towards presenting the Front-end UI of the WaterCPS relevant to Optimization in an indicative, pilot-specific example (Case Dow Bohlen), the Optimization module would request the data shown in Figure 28 from the WaterCPS system through the management internal feature, having been setup through the configuration internal feature for an upcoming Bohlen request optimization. Note that, input data in Figure 28, will be also presented via the Optimization Historic Instance UI, each time the user selects a historical instance from the Optimization History UI (see Figure 27, right).



	COD	TOC	T
cf_1 (mg/L or C)	19.00	6.60	19.80
Cf_2 (mg/L or C)	15.00	5.00	19.50
R_7	0.41	0.47	0.48
R_8	0.49	0.38	0.55
R_9	0.74	0.38	0.42
R_{11}	0.34	0.38	0.35
R_{12}	0.30	0.37	0.30
S_{11}	0.90		
S_{12}	0.89		

cl_{13} (mg/L or C)	13.95	4.35	15.41
cl_{14} (mg/L or C)	13.95	3.90	16.77

Figure 28 Optimization Input received from AquaSPICE WaterCPS based on the pilot specific Templates (indicative values for demonstration) and incorporated in the Optimization Historic Instance UI: parameters of treatment processes per pollutant of interest (COD: Chemical Oxygen Demand, TOC: Total Organic Carbon, T: Temperature) in the Dow Bohlen case.

Figure 28 presents the optimization data for the Dow Bohlen case that the user can access via the Optimization Historic Instance UI. It includes a directed graph (flowgraph) of nodes *V* and arcs *A*, where each arc is linked with a value of flowrate capacity (in volume/time units). Each node is featured with a set of utility-specific parameters. The nodes *i* of Freshwater Intake (*1* and *2*) are linked with a set of contamination factors cf_{pi}, *p* being a pollutant of interest. Treatment processes units (nodes *7, 8, 9, 11, 12*) have a reduction rate value R_{pi} for the contamination factor of pollutant *p*, and S_i for the outcoming flow. Applications nodes (*13* and *14*) have a demand (in volume/time units) and a set of chemical limits cl_{pi}. An indicative example of input data, including three pollutants of interest, is also presented in the accompanying Table. Note that the OptimizeInstance service will solve the optimization problem, with the above input, using an input-specified objective, e.g., for the Dow Bohlen case the objective is the minimization of the incoming flowrate from River Elster (denoted by Node 2 in the flowgraph of Figure 28).

Having completed an optimization cycle with the respective data and for a specific objective, the user receives all the necessary information through the Optimization Response UI in order to take in informed decision about their respective course of action. For instance, as shown in *Figure* 29, the Optimisation Response UI for the Dow Bohlen case provides user information concerning the detailed flowgraph and the contamination factor values for all pollutants of interest. More specifically, each arc of the flowgraph is associated with a flowrate value (in volume/time units), and each node is linked with a set of contamination factor values, indicating the quality of the outcoming flows. The values of nodes *1, 2, 13, 14* were determined by the input data (as they indicate the initial contamination factors and the chemical limits of Applications nodes), as the values of all intermediate nodes are computed by the OptimizeInstance service.

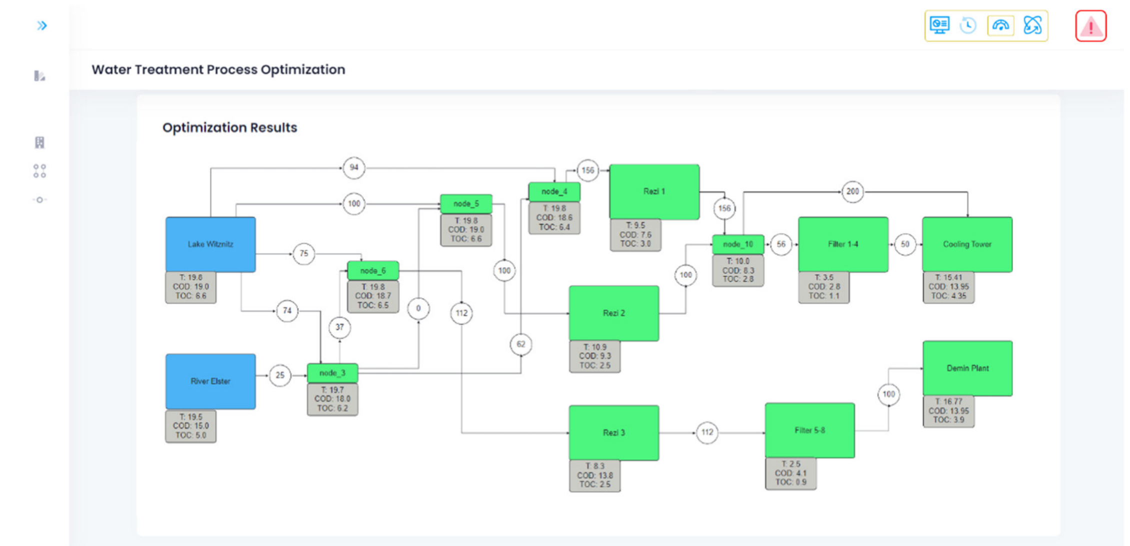


Figure 29 Optimization Response UI

Additionally, once presented to the end user, the optimization solution is stored in its final form and becomes accessible through the Optimization History UI (Figure 27, right) where the user can browse through a list of previously solved instances and select to see the details of one of them. There, the user is presented with the solution provided at that time (see Figure 29) alongside with the configuration values (see Figure 28) the user used to initialize the optimization cycle.

7. Life Cycle Analysis UI

7.1. Indicator Selection and Estimation

The Life Cycle Assessment component of the WaterCPS is tailored to the needs and characteristics of each Case Study. An appropriate list of midpoint impact indicators has been selected during the Case Study Workshops, in agreement with the Case Study leaders and participants. It has been based on the main environmental KPIs of the Case Study and the identified environmental issues. This list cannot be altered by the user of the WaterCPS.

A qualitative Life Cycle Inventory has been prepared for each Case Study, based on the existing and the expected flows of resources and emissions. This Life Cycle Inventory is filled in dynamically based on the input values provided by the Real Time Monitoring system and the Simulation and Modelling component of WaterCPS, leading to the dynamic estimation of the values for these indicators. The characterization factors for these flows have been retrieved from commercial and open-source Life Cycle Assessment databases and from published literature.

The boundaries of the analysis refer to the system depicted in the Process Diagram of the WaterCPS dashboard (left column in Figure 31).

7.2. User Interface Configuration

The WaterCPS user can get an overview of the Life Cycle Assessment results for the entire system in the Dashboard landing page (Figure 30). The current value of the selected dynamic indicators is being displayed. These values can be accompanied by the main contributing factors/flows for each indicator, allowing the WaterCPS user to highlight the main environmental hotspots for each impact category. The indicators displayed here can be customized for each Case Study.

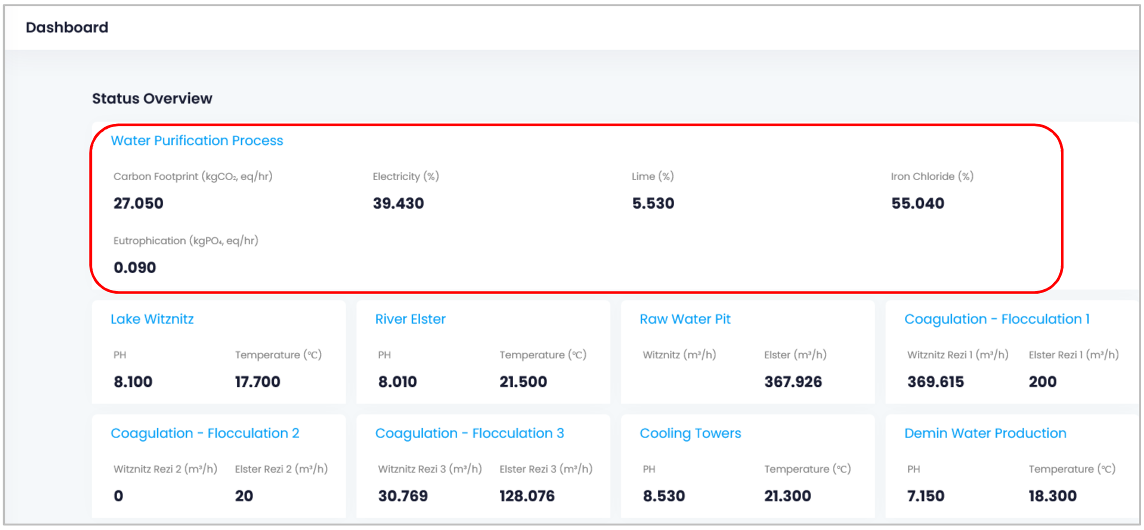


Figure 30. LCA Overview on WaterCPS Dashboard

On the main screen, the user can also get a dynamic overview of the Life Cycle Assessment outputs by clicking on the appropriate tab in the Digital Twin Overview (Figure 31). In this case, apart from the static values of the indicators, the temporal variation of selected indicators for a period of time is illustrated in the corresponding charts.

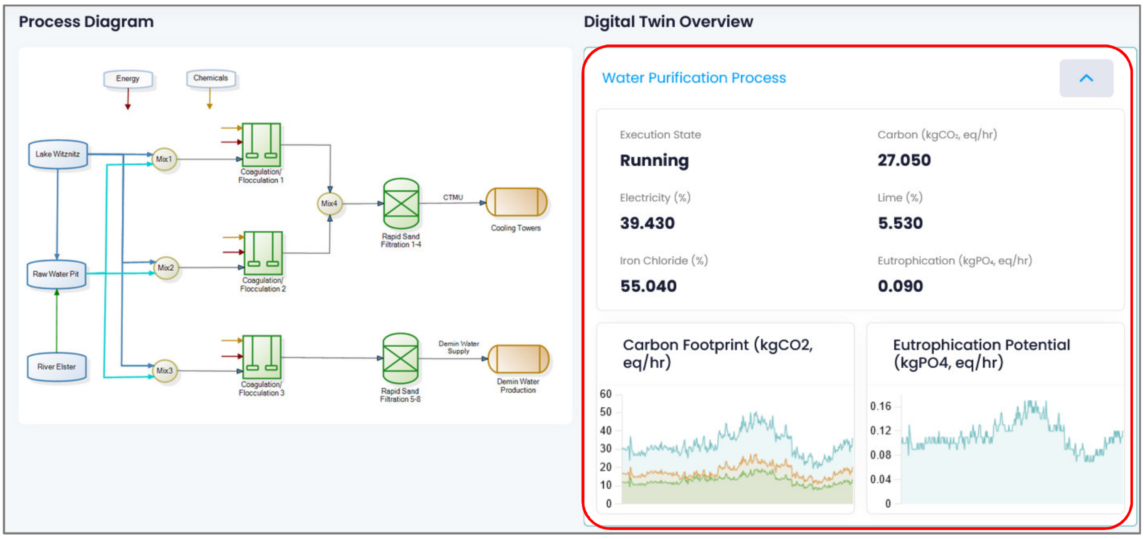


Figure 31. Dynamic overview of the system's Life Cycle Assessment

For each asset of the process or for the entire process, the user can gain access to the LCA monitoring screen, through the top right panel of the corresponding system component (Figure 32). This screen provides a more detailed and dynamic illustration of the indicators, together with the contribution breakdown per flow/resource (indicative chart presented in Figure 33)

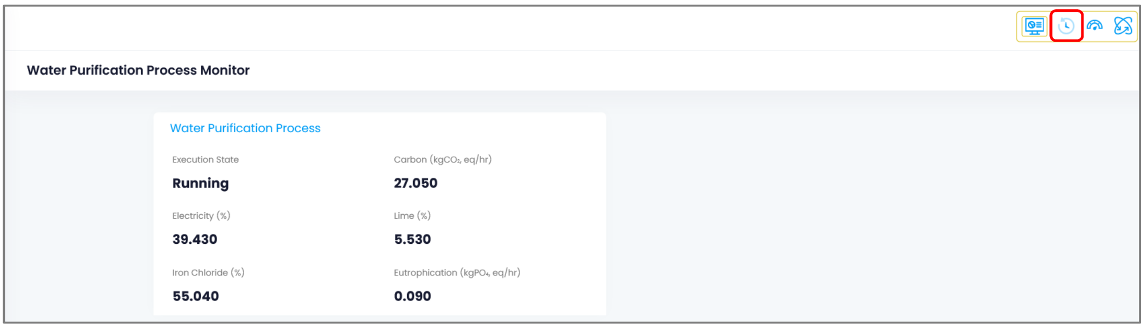


Figure 32. Accessing the Life Cycle Monitoring screen

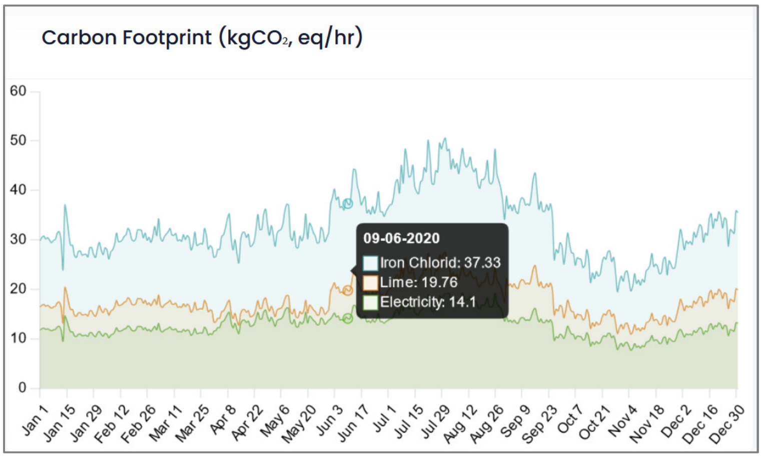


Figure 33. Dynamic illustration of the Global Warming Potential

8. Conclusions and Next Steps

The role of the WaterCPS is to provide a digital virtualisation core (Digital Twin) consisting of a physical system's knowledge representation core as well as process and simulation models. In AquaSPICE, the Digital Twin is being enhanced with smart data analytics, artificial intelligence, optimisation, LCA, and simulation services. An extensible number of tools and applications are currently in development. Based on the specification of WaterCPS User Services, Functional Model and Software Design performed in Task 5.1, this deliverable presented the front-end and human-machine interfaces that provide access to the WaterCPS simulation, optimisation, analytics and LCA functionalities. The interfaces follow the metaphor of a Digital Twin.

Future work includes the integration of the WaterCPS with the IIoT environment, empowered by the RTM platform. The envisaged integration will link physical objects to virtual content and may lead to new requirements for special user interfaces and interaction techniques. Further, deployment of the WaterCPS in the case study environments will clearly raise new requirements with regards to the User interfaces, front-end functionalities and human-machine interactions. We plan to carefully analyse any new requirements pertaining to the integration and deployment tasks and proceed by implementing those deemed of high value to the stakeholders involved in the development and use of WaterCPS.