# iTesla

**Innovative Tools for Electrical System Security within Large Area**

| | | | |
|---|---|---|---|
| **Grant agreement number** | 283012 | **Funding scheme** | Collaborative projects |
| **Start date** | 01.01.2012 | **Duration** | 48 months |
| **Call identifier** | FP7-ENERGY-2011-1 | | |

# Validation Handbook

| **Dissemination level** | | |
|---|---|---|
| **PU** | Public. License: MPLv2. | **X** |
| **TSO** | Restricted to consortium members and TSO members of ENTSO-E (including the Commission Services). | |
| **RE** | Restricted to a group specified by the consortium (including the Commission services). | |
| **CO** | Confidential, only for members of the consortium (including the Commission services). | |

Document Name: Validation Handbook
Work Package: WP7
Task:
Deliverable: -
Responsible Partner:

| Author | | Approval | |
|---|---|---|---|
| Name | Visa | Name | Date |
| | | | |

| DIFFUSION LIST | |
|---|---|
| For action | For information |
| | |

# DOCUMENT HISTORY

| Index | Date | Author(s) | Main modifications |
|---|---|---|---|
| 0.11 | 28.07.2015 | | |
| 0.12 | 20.10.2015 | | |
| 0.13 | 06.01.2016 | | |
| 0.14 | 11.01.2016 | | |
| 0.15 | 01.02.2016 | | Section 4 (many) and Section 6 (small changes) |
| 0.16 | 15.02.2016 | | Section 5.4 and Section 7.11 |
| 0.17 | 25.02.2016 | | Section 4.7 and Section 6 |
| 0.18 | 21.03.2016 | | Section 6.3.12 (IPSO return codes). Diffusion switched to public with MPLv2 license. |

# Table of contents

# Figures

# Glossary

**DB**: Data Base
**CIM:** Common Information Model – data exchange format agreed internationally among transmission system operators.
**SN**: Snapshot
**DACF**: Day Ahead Congestion Forecast
**IDCF**: Intra Day Congestion Forecast
**FO**: Forecast (includes DACF and IDCF files)
**IIDM XML:** data exchange format internal to the project iTesla. IIDM means Internal iTesla Data Model.
**MCLA:** “Monte Carlo-Like” Approach
**OVH**: iTesla hosting service provider
**SSH**: Secure Shell - protocol for secure network communications
**WCA:** Worst Case Approach
**OPF:** Optimal Power Flow
**FPF:** Fuzzy Power Flow

# 1. EXECUTIVE SUMMARY

This handbook aims to facilitate the validation of the iTesla toolbox.

It compiles commands made available and describes the validation method. This document is meant to evolve regularly along with the evolutions of the platform.

So far:

- The validation methodology is described for a 7-bus test network and the French network
- The offline part is fully implemented (except for the importance sampling part)
- The online part is now available to the validators
- Only Eurostag is integrated (not Dymola)

# 2. Technical generalities:

## 2.1. Test platform generalities:

The test platform is hosted by OVH. It consists of three Linux Centos 6 servers with 16 cores each. It can be accessed through SSH. The master node (ns6375578.ovh.net) is used for command line interactions.

For test purposes, 2 other servers were made available: ns6375579.ovh.net and ns6375582.ovh.net.

The HPC platform is too complex to be used by validators for now.

For now, the online part is available on servers ns6375578.ovh.net and ns6375579.ovh.net.

## 2.2. Description of the general architecture:

The general architecture of iTesla from a functional point of view is represented below. It was described in detail in deliverable D1.2.

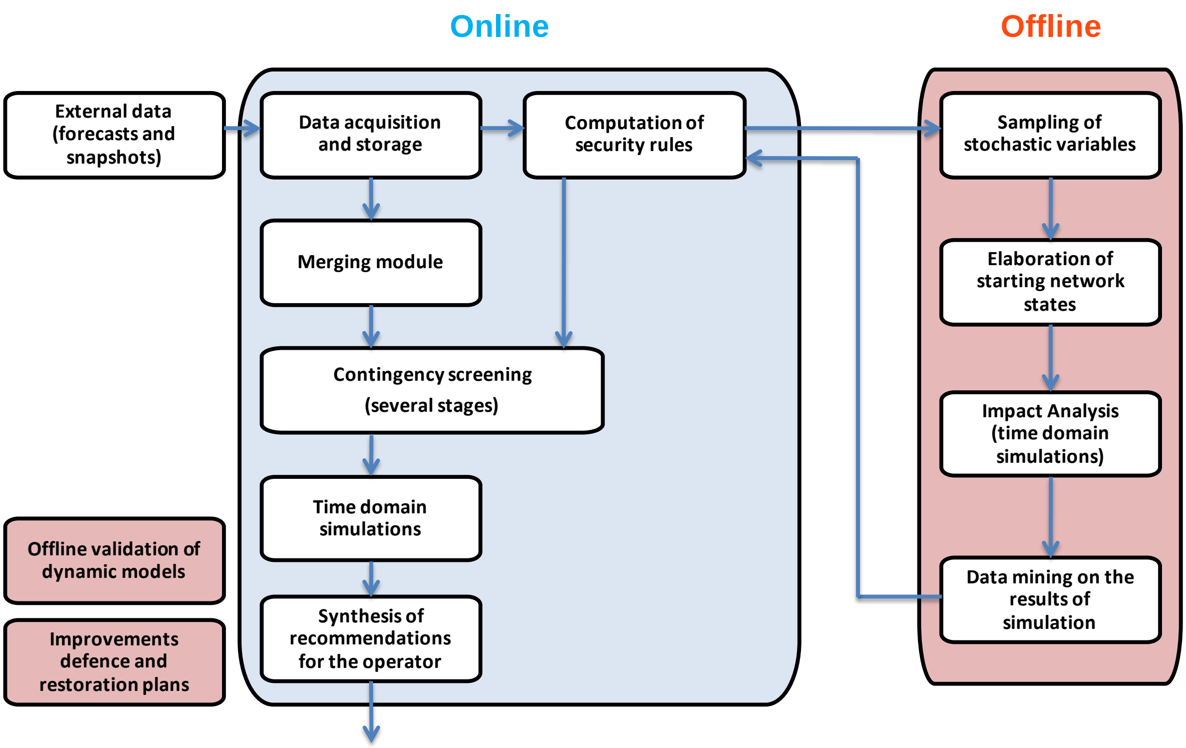


Figure 2 - 1: General architecture (functional)

The parallel between the general architecture and its implementation as a computer tool will be made. Throughout the rest of the document, the different functions or blocks forming part of the whole architecture will be named after the associated work package.

## 2.3. WP4 architecture:

The WP4 (or offline part) computer architecture can be represented that way:

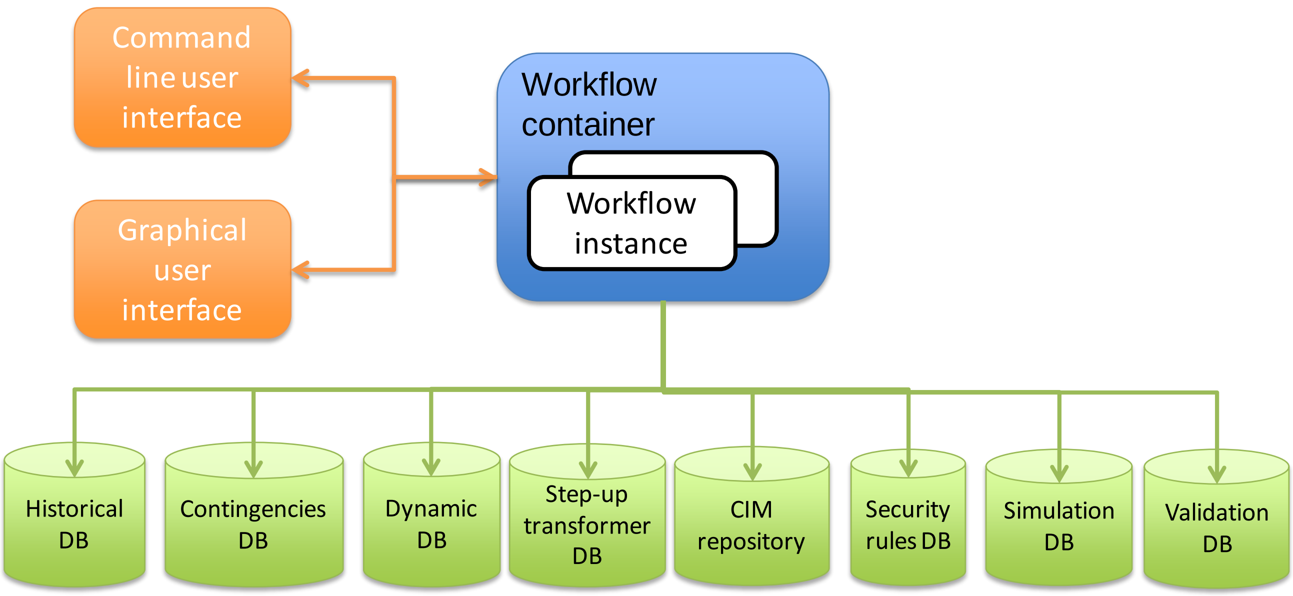


Figure 2 - 2: WP4 general computer architecture

The user interacts with the platform through command lines or a graphical interface. For validation purposes, command lines are (for now) more flexible and more powerful in terms of available functionalities. Some of the command lines contain instructions to export data in user friendly formats (csv, xml, eurostag, mat, ampl etc.) to be processed as the person in charge of the validation decides to.

It is not possible for now to run several workflows in parallel but it will be possible in the future. Several DB contain the data required at the different stages of the workflow (intermediate and final results). The content of the various DB and the configuration files associated will be detailed in annex.

The scheme below illustrates the interactions between the different steps of WP4, their inputs, outputs and the associated data storage.

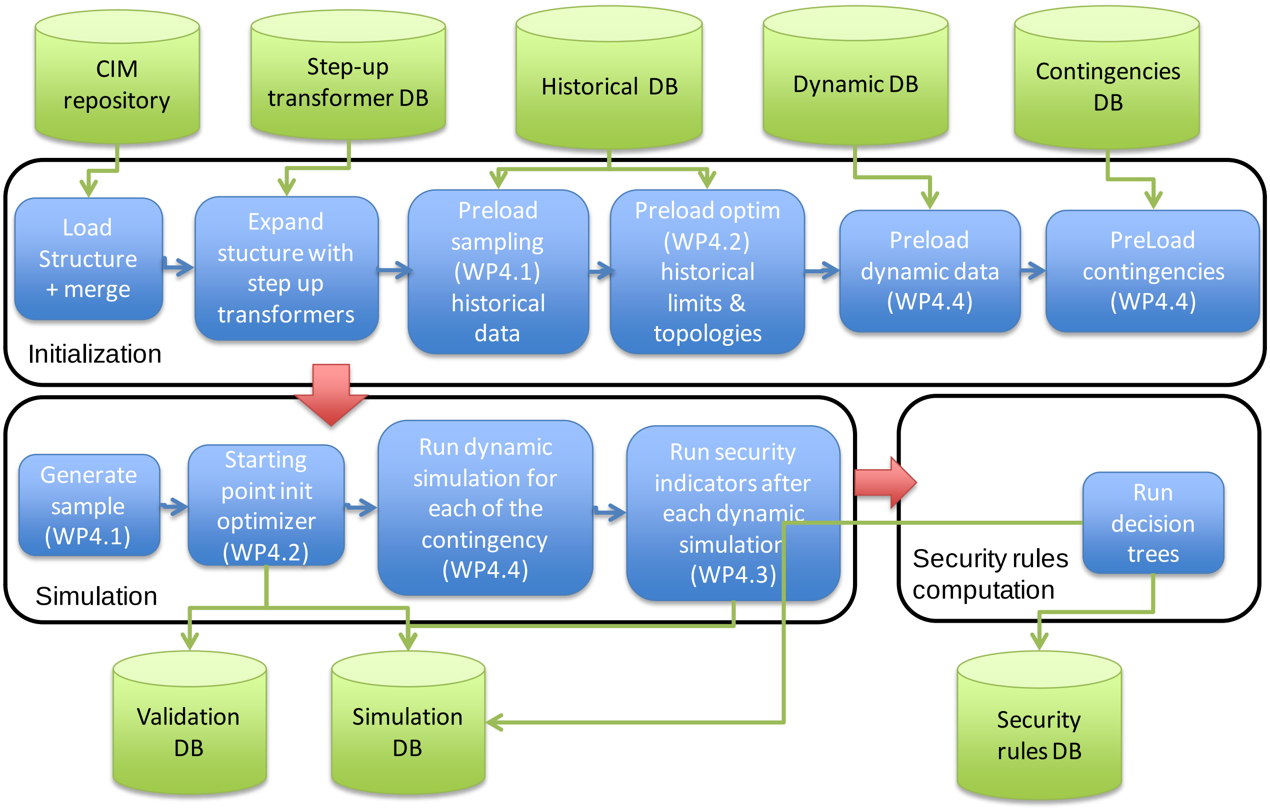


Figure 2 - 3: WP4 detailed computer architecture

## 2.4. WP5 architecture:

The WP5 (or online part) computer architecture can be represented that way:

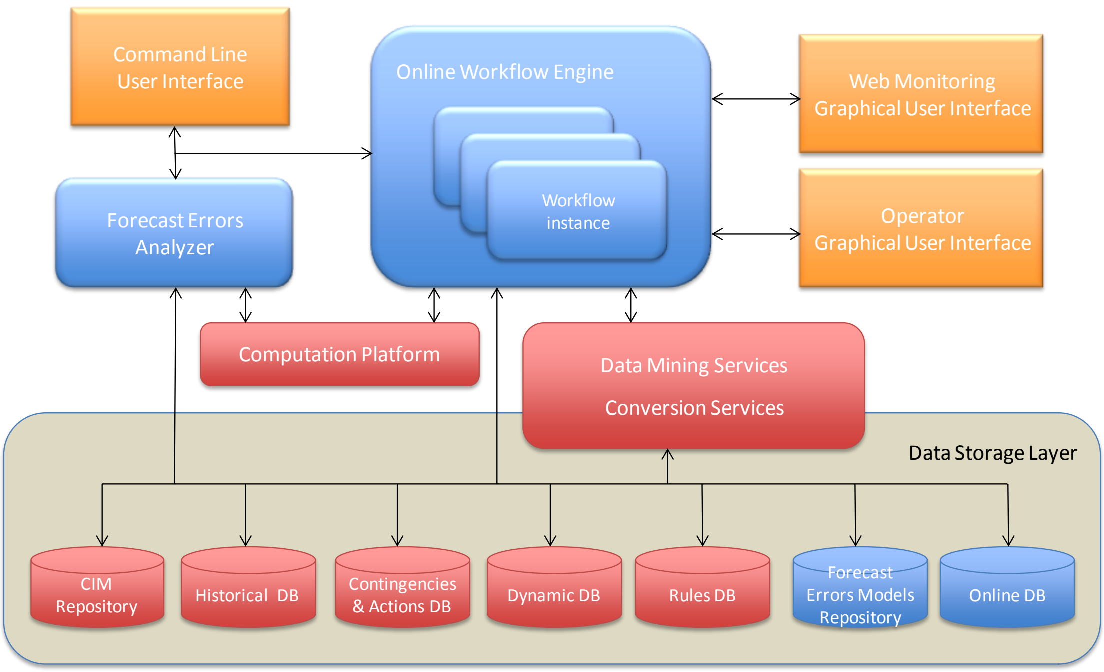


Figure 2 - 4: WP5 general computer architecture

The scheme below represents one workflow instance for WP5:

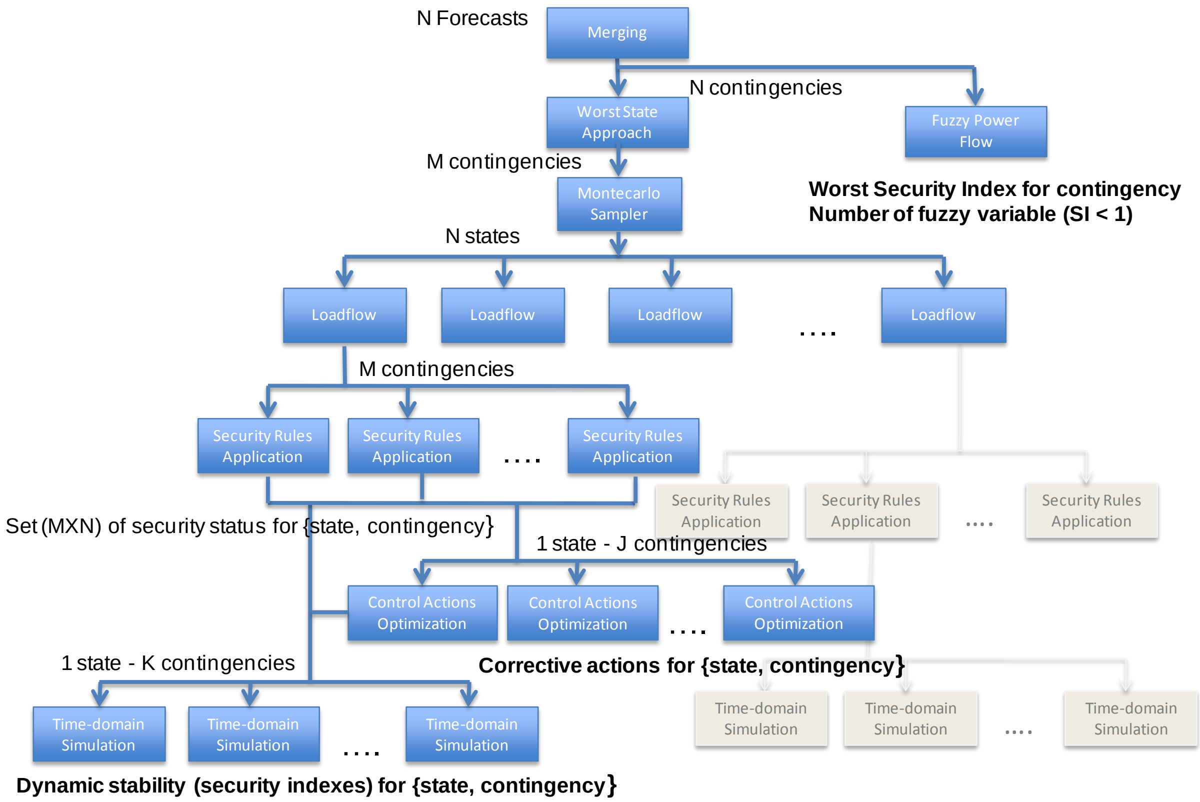


Figure 2 - 5: Workflow instance for WP5

The scheme below illustrates the interactions between the different steps of WP5 and the data storage.

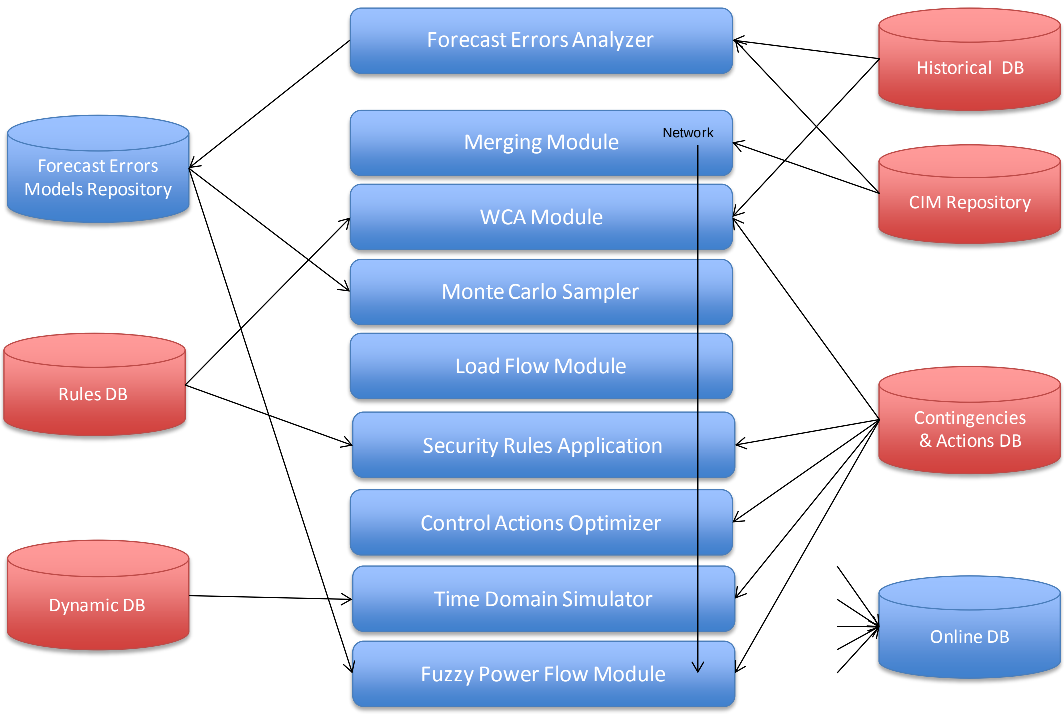


Figure 2 - 6: Interaction with the databases

The online database is connected to (can be used by) all the online modules but the forecast errors analyzer.

# 3. OFFLINE VALIDATION METHODOLOGY:

2 "levels" of tests can be run:

- on a small fictitious test network (the 7-bus test network described in section 3.1);
- on the different real power system use cases defined for WP7.

The first tests on a real power system were performed on the French network (on the 400 kV and 225 kV voltage levels).

Two approaches were defined for the offline validation methodology:
1. Step-by-step validation of each software module, aiming to validate the accuracy of the results produced by each module (described in section 3.2);
2. Overall WP4 validation, aiming to evaluate the accuracy of the decision trees (DTs) selected to be used online (described in section 3.3).

## 3.1. Elaboration of a small test network

A 7-bus network was implemented for tests. The range of tests which can be performed with it is limited but it is useful for debugging, basic tests and training purposes.

The network is represented on the diagram below.

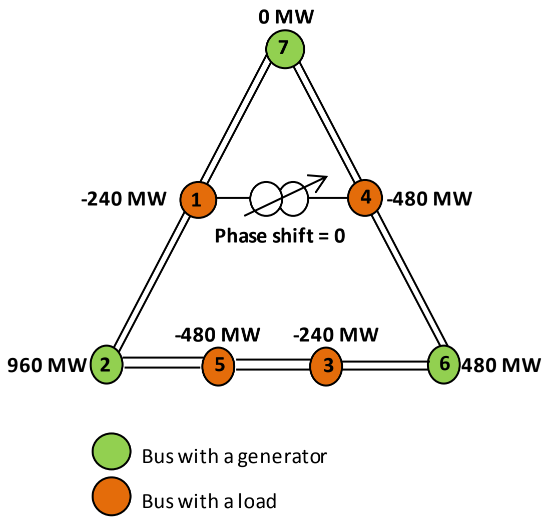


Figure 3 - 1: 7-bus test network

The test network in its genuine version is very stable. Variants of this network were created keeping the same network structure and adjusting input parameters (in the network .CIM file, in the dynamic data or in the platform configuration) to bring the network to its limits with respects to one specific criterion after one specific fault.

The parameters were adjusted for an average load situation (240 MW for buses 1 and 3, 480 MW for buses 4 and 5). For validation purposes, historical data for the stochastic variables scattered around the average value were generated to tip the balance towards a stable or unstable state depending on their values.

These historical data were integrated in the CIM base to generate one CIM file per generated network situation (all these files are the historical database). One drawback of this method is that the network situations are not necessarily consistent: the flows on the lines, the voltages and the production of the centralized power plants are identical to the ones in the base case.

The benefits of working on a small network are:

- A better understanding of the outputs
- A low calculation duration

So far, 5 variants have been implemented. The methodology to adjust parameters for each variant is described below.

### 3.1.1. Transient stability

The duration of a chosen fault was adjusted in order to reach the limits of synchronism on a specific generator. The fault duration had to be changed in the platform parameters. The .CIM file and the dynamic data files remain unchanged.

Apart from the fault duration and location, the stability of a generator is affected by many parameters. One of them is the active and reactive power produced by the generator which is affected by the load.

When different load situations are simulated, the stability of the generator is affected (the starting point of the generator on the diagram is different on each situation). Some situations lead to a loss of synchronism and others not.

This variant proved to be very difficult to use because the speed reference (used in the indexes) is the weighted average of all the speeds of the synchronous generators. On a small network like the 7-bus network, when a fault occurs, all the generator speeds and internal angles change simultaneously and very long fault durations are needed for one generator to lose synchronism. These fault durations triggered calculation failures in Eurostag.

Transient stability indexes are now validated with the French network.

### 3.1.2. Small-signal stability

The dynamic parameters of the generators in the dynamic data files and impedances in the .CIM file were changed in order to generate inter area oscillations between two of the generators. The damping ratio was calculated for the average load configuration. The damping ratio limit of the index is set to this value.

This small system was analyzed and tuned thank to SMAS3 (Selective Modal Analysis of Small Signal Stability) software which is a state of the art computer program package for the study of small signal stability in electric power systems.

Similarly to transient stability, when different load situations are simulated, the stability of the generator is affected leading to situations where the damping ratio limit is violated or not.

This variant can still be useful to run tests on the small signal stability index.

### 3.1.3. Overload

Maximal admissible currents were adjusted to the limit in chosen lines after a given fault. The appropriate parameters were changed in the input .CIM file. The dynamic files remained unchanged.

Currents are affected by the different load situations leading to overload situations in some cases.

This variant was used for many tests and presentations. It is particularly useful for training.

### 3.1.4. Overvoltage

Impedances of the lines were adjusted in order to reach an upper voltage limit after a given fault. The appropriate parameters were changed in the input .CIM file. The dynamic data files remained unchanged.

Voltage is affected by the different load situations leading to overvoltage situations in some cases.

This variant was difficult to use because of the lack of historical data for the voltages making the initialization of states (WP4.2) difficult (no quantile calculation).

### 3.1.5. Undervoltage

Impedances of the lines were adjusted in order to reach a lower voltage limit after a given fault. The appropriate parameters were changed in the input .CIM file. The dynamic data files remained unchanged.

Voltage is affected by the different load situations leading to undervoltage situations in some cases.

This variant was difficult to use because of the lack of historical data for the voltages making the initialization of states (WP4.2) difficult (no quantile calculation).

### 3.1.6. Synthesis

The figure below summarizes what is related to the variants:

| Variant name | Changes | Index tested |
|---|---|---|
| smallsignal | regulations, r, x… | SMALLSIGNAL |
| synchroloss | fault duration and location | TSO_SYNCHROLOSS, TRANSIENT |
| overvoltage | r, x, voltage limits | TSO_OVERVOLTAGE |
| undervoltage | r, x, voltage limits | TSO_UNDERVOLTAGE |
| overload | current limits | TSO_OVERLOAD, OVERLOAD |

Figure 3 - 2: Description of the 7-bus variants

The voltage stability index has not been implemented in the platform yet so will not be tested in a first step.

#### 3.1.7. Implementation in the platform

Although all the commands described in chapter 5 apply to the 7-bus network, it may be useful to describe how the 7-bus network was implemented in the platform in order to conduct the tests properly. The technical specificities of the 7-bus network are described in annex 7.11.

### 3.2. Step by step validation

The blocks constituting the offline part are to be checked one by one. The main steps of this validation procedure are described next.

#### 3.2.1. Characterize the general workflow results obtained before DT computation

After running a workflow from WP4.1 to WP4.4 (i.e. before decision trees computation), perform the following:

- Evaluate the number of 'OK', 'NOK' (i.e. crashes) states for each step of the offline process (from WP4.1 to WP4.4), by analyzing the contents of the obtained offlinedb.csv file.

- Evaluate the rate of success of each step of the workflow to generate network states by computing: $\boldsymbol{Success\ Rate}_{step} = (\#(\boldsymbol{OK}\ \text{states})_{step}/\#(\boldsymbol{OK}\ \text{states})_{step-1}).\mathbf{100}\%$

- Evaluate the rate of success of the overall workflow to generate network states by computing: $\boldsymbol{Success\ Rate}_{final} = (\#(\boldsymbol{OK}\ \text{states})_{final}/\#(\boldsymbol{sampled\ states}\ \text{in WP4.1})).\mathbf{100}\%$

- Evaluate the number of generated stable and unstable states for each contingency/security index pair.

### 3.2.2. WP4.1 (sampling of stochastic variables)

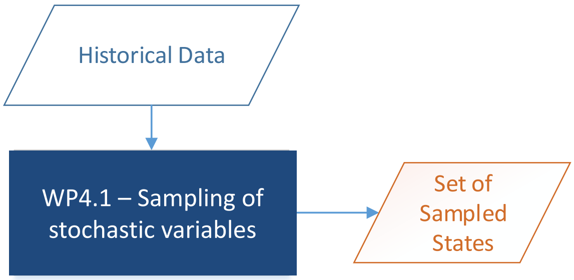


Figure 3 - 3: Sampling of stochastic variables

- Validate the sampling quality by checking consistency of the generated sampled stochastic variables with the input SN historical data:
  - Run Imperial college validation module[1] on relevant sets of stochastic variables to check that historical and sampled data present similar statistical attributes (this tool mainly provides visual comparison between historical and sampled data and analysis of some statistical indicators).
  - Other tests can be performed using the raw historical and sampled data extracted thanks to the same module.
- Some additional test procedure can be performed for WP7 use cases (not valid for the small test system, since it is a fictitious use case), namely the following:
  - When running Imperial college validation module on relevant sets of variables, check correlation for chosen subset of variables.
  - Check consistency using special knowledge on behavior by zones, production type, etc.
  - Analyze random loads (correlations, probability densities, phase shift, etc.)
  - Analyze particular points (specificities familiar to operators).

### 3.2.3. WP4.2 (starting points)

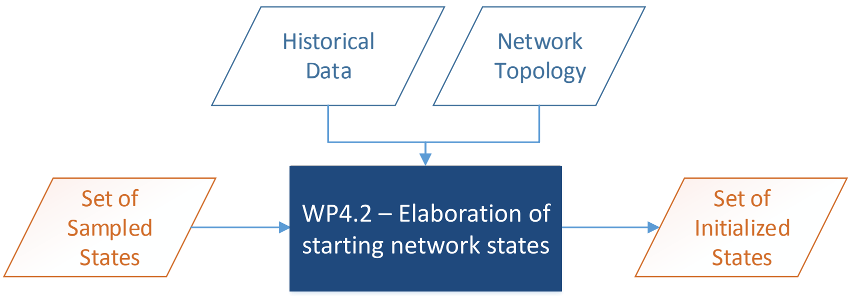


Figure 3 - 4: Elaboration of starting network states

- List cases of divergence and analyze their origins with WP4.2 experts using the metrics.csv file. Focus on non-extreme sampled cases.
- Analyze the accuracy of steady-state operating conditions, which requires expertise on the system. For this analysis, initialized states can be visualized on the offlinedb.csv file. Besides, selected states can be exported to input data of a dynamic simulator (Eurostag, dymola…) for analysis. Examples of operating conditions that can be checked are:

[1] See full description in annex 7.12

- Production repartition.
- Phase shift transformer taps.
- Voltage references.
- Switched on capacitors/inductors (not valid for the small test system).
- Topology choice consistency (not valid for the small test system).
- If U/Q/P static domain of generating units is compatible with the dynamic simulation data (this may have a strong impact on the quality/feasibility of the dynamic simulations performed for each initialized state).

### 3.2.4. WP4.4 (time domain simulation)

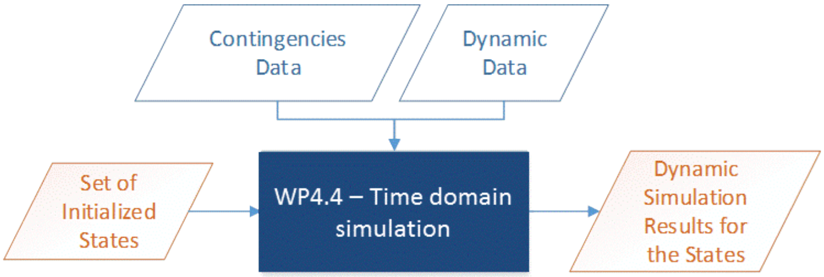


Figure 3 - 5: Time domain simulation

- Analyze “crashes” in dynamic simulation.
- If necessary, export selected initialized states to input data of a dynamic simulator (Eurostag, dymola…) for analysis.
- Check accuracy of event sequences.

### 3.2.5. WP4.3 (security indices computation)

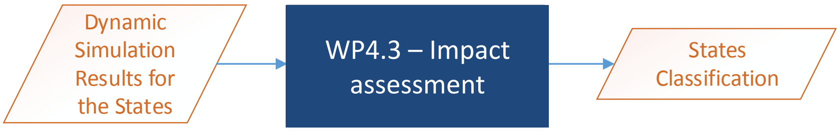


Figure 3 - 6: Security indices computation

- Compare the computed indices with the results of dynamic simulations (for a selected number of samples):
  - Run again picked simulations (randomly or according to the obtained security classification results) in a dynamic simulator (Eurostag, dymola…) manually and confront the results with WP4.3 outputs (i.e., with the security indices results that are described in the offlinedb.csv file and some other detailed information about the violated operating conditions that are described in specific debug files).
- For misclassified samples: check parameters of the module and, if necessary, report to developers.

### 3.2.6. WP4.5 (decision trees computation)

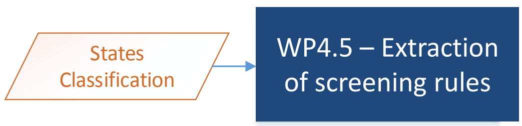


Figure 3 - 7: Extraction of screening rules

- <u>For each security problem (contingency/security index couple):</u>

a) Compute alternative DTs with "DataMaestro" software by using different values for the training parameters (namely for the alpha, trueThreshold and maxSplit parameters). The k-fold Cross validation method must be activated. As an alternative, the iTesla toolbox also enables the computation of DTs with "Matlab".

b) Evaluate the test set accuracy of each computed DT, by analyzing the following indicators:
   - *#ucu*: Number of Unstable states classified Unstable
   - *#scs*: Number of Stable states classified Stable
   - *#ucs*: Number of Unstable sates classified Stable (Number of missed alarms)
   - *#scu*: Number of Stable sates classified Unstable (Number of false alarms)
   - *(#ucs + #scu)/(#states) x 100%*: Global Classification Error
   - *#ucs/(#unstable states)*: Missed Alarm Error
   - *#scu/(#stable states)*: False Alarm Error
   - *#ucs/(#ucs+#scs)*: Probability of a case is unstable while classified stable
   - *(#ucs+#scs)/(#states)*: Probability of a state being classified stable

c) Select the best DT (for the WCSA and for the MCLA), by pursuing the following criteria:
   - Probability of a case is unstable while classified stable < 5%, to obtain a minimal missed alarm error;
   - Max {Probability of a state being classified stable}, to increase the filtering capability of the online part;
   - The test set errors are similar to the training set errors, to avoid over-fitting or under-fitting.

- For each selected DT:

a) Perform an exploratory data analysis of the stability domain by plotting the stable and unstable states that were correctly and incorrectly classified by the DT in a 2 - dimensional space given by:
   - principal component analysis[2].
   - the predictor variables identified by the DT[3].

   This analysis may give valuable information about the dispersion, convexity and representativeness of the data set used for DT computation.

b) Check accuracy of the DT with the historical data that was used for DT computation:
   - Run a dynamic simulation impact analysis (i.e. run WP4.4 and WP4.3) for all the SN of the historical data that was used for DT computation and then compare the classifications provided by dynamic simulation with the classifications provided by the DT. Analyze the usual statistics to evaluate the generalization capability of the DT[4].

c) Check consistency between the sampled and historical data used for DT computation:

[2] Functionality provided by a developed Matlab interface described in annex to be written…
[3] Functionality provided by a developed Matlab interface described in annex to be written…
[4] Functionality provided by a developed Matlab interface described in annex to be written…

- Obtain other statistical comparison between the historical and sampled data used for DT computation (like the distribution of stable/unstable states for the samples and the historical data, for the variable selected to split the tree, at each splitting test) [5]. The two types of data must present similar statistical attributes.

## 3.3. WP4 Overall Validation

The main objective of this global validation is to assess the generalization capability of the DTs, selected to be used online, with more realistic conditions, namely by evaluating the accuracy of the DTs:

- with historical data not used for training;
- for the states immediately preceding a recorded severe contingency (only for the DTs that consider the same contingency).

The WP4 overall validation procedure comprises the following steps:

a) Separate the available historical data (SN) into two sets: **Set A** that will be used to sample states and create the DTs and **Set B** that will be used as an independent test set containing historical data not used for training. Set A must contain SN that occurred before the SN of Set B. If possible, Set B should contain chronologically connected SN collected just before of the contingency under analysis that lead to the security issue of the underlying use case.

b) For the contingency and security problem under analysis, execute all WP4 commands before DT computation. If necessary, check offlinedb.csv file to detect possible inconsistencies in the modules output and to make sure that a representative set of data exists for the DT computation.

c) For each security problem (contingency/security index couple), perform a fine tune the DT parameters by following the training procedure already described in section 3.2.6.

d) <u>Check the generalization capability of the computed DTs with **Set B** historical data, not used for DT computation</u>: Run a dynamic simulation impact analysis for all the SN of the selected historical data and then compare the classifications provided by dynamic simulation with the classifications provided by the DT. Infer the usual statistics to evaluate the generalization capability of the DT.

<u>In the special case of the 7 bus use case</u>, there are no consistent network states in the historical CIM data base, only copies of the base case in which the loads were replaced with the generated “historical” load data. It is then hardly possible to create a big test set from the historical data base. Nevertheless, several options are possible:

- Test the security rules on the base case
- Create a few new CIM files from the base case changing manually the load values and the production values (using Convergence for example).

[5] Functionality provided by a developed Matlab interface described in annex to be written…

# 4. ONLINE VALIDATION METHODOLOGY:

The online platform consists of several sub-modules, as defined in D5.2 and whose implementation is described in D5.3. Further enhancements are presented in D5.4.

The validation of the online platform consists of different steps:

1) Checking of the operation of each module integrated into iTESLA against its functional specifications. This part includes verifying the consistency of the data porting from IIDM format (i.e. iTESLA format) to the format of the modules, and the consistency between models. It also implies to check module results against suitably defined benchmarks, in order to validate the functional specifications themselves.
2) Testing of the online workflow operation.
3) Evaluating the performances of the individual modules.
4) Evaluating the overall online workflow performances.

Task 1) was carried out and reported in D5.4. Task 2) was partially carried out within WP5.3 and WP5.4, and it will be completed in WP7. Tasks 3)-4) are specifically targeted in this document.

The different stages of validation may be carried out with different sets of input data, from simpler to more complex:

a) A small input data set (7-bus network), consisting of “controlled” (i.e. known, purposely defined) data;
b) A small input data set (portion of a real grid), consisting of real data (e.g. real forecasts and snapshots);
c) A real case data set (full grid).

Before defining the validation criteria of the online platform, this section starts with introducing the new concepts for security assessment in the iTESLA perspective. Following, the benchmark for security assessment is proposed. Performance indices for validation are then defined, for a single forecast and for a set of forecasts. Procedures for validation of the modules are then described. Indices to evaluate the quality of the validation set are also proposed.

In order to better understand the set of intermediate outputs produced by the different modules in the validation mode, a scheme providing an overall picture of the online workflow is provided, in which the specific outputs are described.

## 4.1. Security criteria under uncertain domains

Compared to conventional security assessment, iTESLA has the additional feature of considering uncertainties in dynamic analyses. This requires to **extend the security criterion definition to uncertain domains**.

The following definitions are proposed.

**Conventional security analyses**:
The **system state under analysis** is secure with respect to a contingency, if no security index exceeds defined thresholds in the response of the system to the contingency, starting from the considered state.

**Security analyses with uncertain domains** (i.e. as with iTESLA):

> The state under analysis (**base case** forecast) is secure with respect to a contingency, if no security index exceeds defined thresholds in the response of the system to the contingency, starting from the considered base case **and from any state within the uncertainty domain of the base case**.

In practice, when considering uncertainties, security must be assured in the whole region of uncertainty. The issue is of course to define this region:

> Suitable **methods and criteria to define the uncertainty region** must be defined.

## 4.2. Benchmark for security assessment under uncertainty

Before introducing performance indices for the iTESLA online assessment, a reference (benchmark) procedure must be defined, that implements the security criterion for uncertain domains, stated above.

The following procedure is proposed to analyze the security of an uncertain forecast state with respect to a contingency:

1. Starting from the base case, **sample** a (significant) **set of states** by MCLA.
2. Perform detailed **T-D simulation** of the contingency applied to the base case and to the samples.
3. Compute the **security indices** for all T-D simulation results (i.e. the same ones used for security rule training).
4. Evaluate the security of the contingency with respect to the base case and to each sample, by applying the security judgement criterion relevant to each security index (i.e. comparison of the index with predefined threshold).
5. If **at least one** security index for **one** of the above simulations is beyond the relevant security threshold, the **case under analysis** is unsecure with respect to the considered contingency; otherwise it is secure.

The next figure provides a scheme of the computation procedure, that can be repeated for the whole set of contingencies under analysis. In particular, the detail of the different phenomena, caught by different security indices, is shown.

The following remarks apply:

- The uncertainty domain model computed in MCLA is assumed as the “most complete” available one. It has been validated in WP5.
- The security indices have been validated in WP4.
- T-D simulation is assumed as representing the «real» system behavior.
- If the sampling is “large enough”, then it may cover the uncertainty domain. However, as the number of stochastic variables increases, as it is with real cases, it is more and more difficult that the dimensions of the uncertainty domain are explored exhaustively.

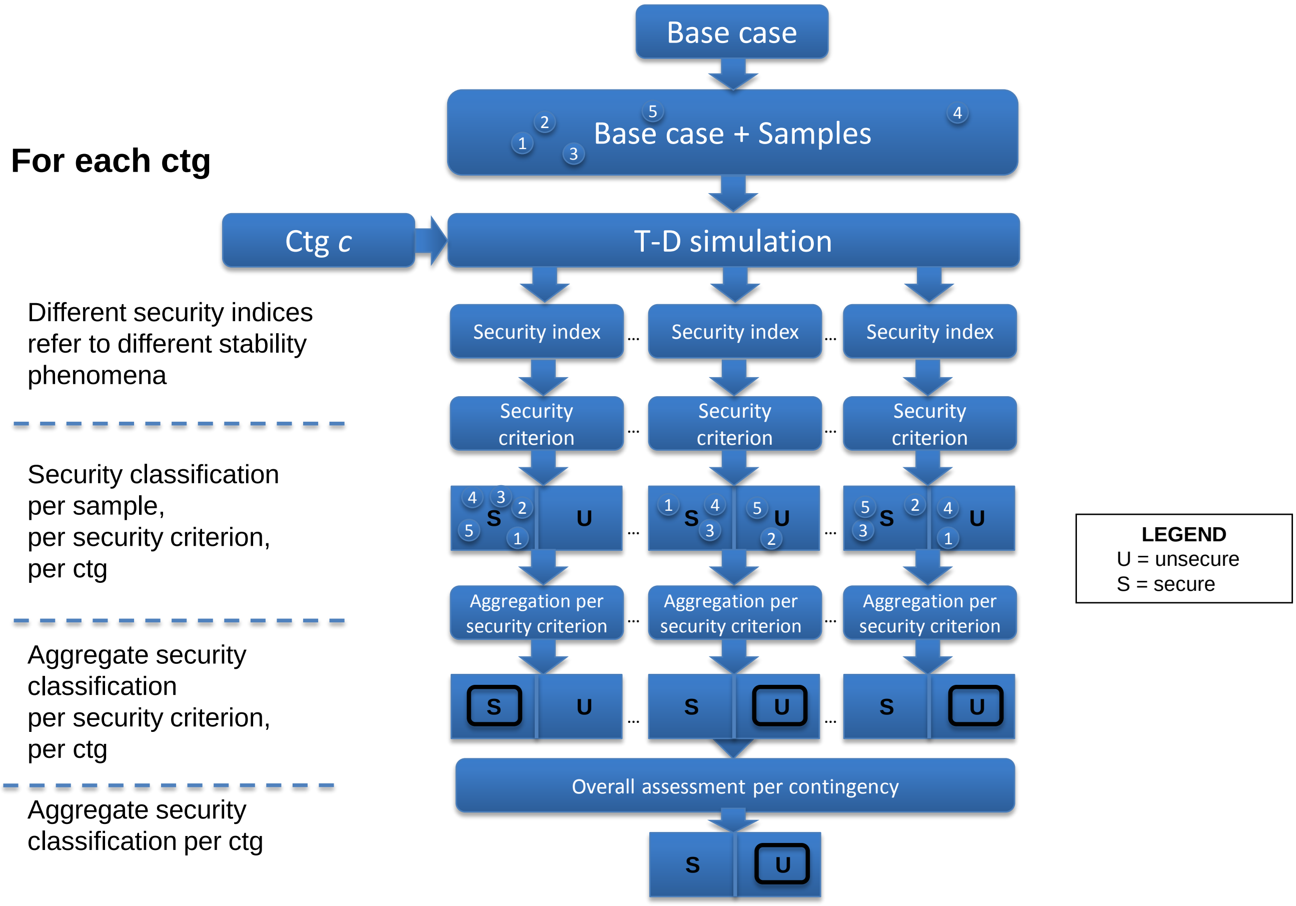


Figure 4 - 1: Benchmark calculation procedure

## 4.3. Performance indices for validation

As recalled above, the online platform consists of modules for merging, filtering, control, detailed assessment, fuzzy power flow, and GUI. Performance criteria for the computational modules (i.e. all modules except the GUI) regard:

(1) the quality of the judgement, i.e. the ability to distinguish between

- **correct judgement**;
- **missed alarms**;
- **false alarms**.

**Missed alarms** refer to contingencies that are judged secure by the assessment process (namely iTESLA) but are classified insecure by the benchmark assessment approach.

**False alarms** refer to contingencies that are classified as insecure but are actually found secure.

These concepts are sketched in the next figure.

(2) the speed of the evaluations:

- **Computation time**.

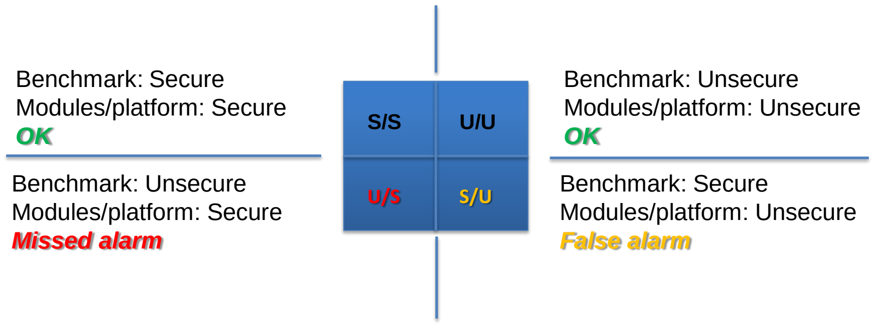


Figure 4 - 2: Comparison between benchmark and filtering/assessment results

The approach to evaluate the above performances is based on the comparison between the **benchmark** and the **modules output.** In particular, as far as the quality is concerned, the following classes of indices are introduced:

- **Accuracy**;
- **Efficiency**.

which have to do with missed and false alarms, respectively.

The indices can be defined:

- for individual base cases (individual forecasts);
- in an aggregated way, for a set of forecasts.

In particular, the second class is important to report about the generality of the performance level of the classifications.

### 4.3.1. Indices applying to a single base case

The following basic indices apply to the results of analysis of individual forecast states.

**Accuracy**

By accuracy we indicate the guarantee that no unsecure contingency is discarded. This property should have the highest priority in filtering methods, i.e. it is mandatory that dangerous contingencies are retained for detailed analysis and/or operator alert.

Indices are defined first for a single base case, then for a generic set of base cases. The contingency list is assumed to be fixed and sufficiently exhaustive.

- *Accuracy* = [1 – ( #Missed_alarm_ctgs / #unsecure_ctgs ) ] x 100

This definition applies to a specific forecast (base case) and contingency list.

The number of unsecure contingencies (#unsecure_ctgs) is calculated from the benchmark procedure.

The number of missed alarms (#Missed_alarm_ctgs) is computed by comparing the classification outcome (secure/not secure) of the module under analysis (or the chain of modules) and the outcome of the benchmark, according to the definition provided above.

**Efficiency**

By efficiency we refer to the goodness at correctly identifying secure contingencies. This property characterizes the ability of the method to discard harmless contingencies, therefore it is a crucial property for filtering methods. However, its priority is smaller than accuracy. Efficiency can be defined for a single base case over a set of contingencies as:

- *Efficiency* = [1 – ( #False_alarm_ctgs / #secure_ctgs ) ] x 100

Again, this definition applies to a specific forecast (base case) and contingency list. The number of false alarms and secure contingencies are computed similarly to what reported in the previous subsection.

**Time**

The time used for the filtering process obviously depends on the platform used. Given the platform, an average computation time can be simply defined as:

- $t_{average} = t_{Total\ computation} / \#ctg$

In general, as the computation time required for unsecure contingencies may substantially differ from that of secure ones (according to the specificities of filtering method), the average time for unstable and stable contingencies may be distinguished:

- $t_{average\_secure} = t_{Total\ computation\ relevant\ to\ secure\ ctg} / n_{secure\ ctg}$
- $t_{average\_unsecure} = t_{Total\ computation\ relevant\ to\ unsecure\ ctg} / n_{unsecure\ ctg}$

### 4.3.2. Indices applying to a set of base cases

After running the platform over a series of base cases, **global indices** can be computed as:

1. **Mean** of individual indices over the number of base cases:

   GlobalPerformance =
   = $\Sigma_{i=1}^{n}$ Performance_index$_{base\ case\ i}$ / $n$

   where

   Performance_index = *accuracy*, *efficiency* or *time* as defined in the previous subsection
   $n$ = # of base cases considered

2. **Variance** of the individual indices above

### 4.3.3. Use of the indices

Performance indices can be applied to both:
1) **individual modules (WCA and MCLA)**, and
2) **the overall online filtering/assessment process**.

As far as the second application is concerned, the classification results are those obtained running the whole online platform, in particular the filtering chain. For example, in normal operation of the platform, contingencies discarded by WCA are not analyzed by MCLA any longer. This may affect overall performances results.

### 4.3.4. Additional indices for MCLA

With MCLA, accuracy and efficiency can also be defined *for each phenomenon* (described by rules trained on specific security indices). The relevant indices can be defined as:

- *Accuracy* = $(1 - \#Missed\_alarms\_ctgs_{specific\_phenomenon} / unsecure\_ctgs_{specific\_phenomenon}) \times 100$
- *Efficiency* = $(1 - \#False\_alarm\_ctgs_{specific\_phenomenon} / \#secure\_ctgs_{specific\_phenomenon}) \times 100$

## 4.4. Procedure for validation of WCA

Validation of WCA may be split into two stages:

1) Validation of the **uncertainty domain** against MCLA domain**;**
2) Validation of the **security classification** against the benchmark.

Moreover, an overall approach can be introduced, as:

Validation of WCA **security classification** against MCLA security classification

In the following, all three approaches are described, though only the third one will be implemented during the project.

### 4.4.1. Validation of the uncertainty domain against MCLA domain

The uncertainty domain computed by WCA (see D5.4 for details) can be compared against the one computed by MCLA.

Because the WCA domain is expected to be larger than the one of MCLA, a possible approach to validation may consist of checking if the samples extracted by MCLA are contained in the domain of WCA.

1. Consider a **base case** *i*
2. Sample a set of $\#MCLAsamples_{base\_case\_i}$ states by MCLA
3. Set $\#MCLAsamples_{inside\_WCAdomain\_base\_case\_i}$ equal to zero
4. For each sample:
   - check if it is within the uncertainty domain defined by WCA (this task is easy, as the WCA domain is defined by a set of linear constraints);
   - if yes, increment by one the variable $\#MCLAsamples_{inside\_WCAdomain\_base\_case\_i}$
5. Calculate performance index relevant to base case *i*:

$$Performance_{uncertainty\ domain}(base\_case_i) =$$
$$= \#MCLAsamples_{inside\_WCAdomain\_base\_case\_i} / \#MCLAsamples_{base\_case\_i} \leq 1$$

If this index is equal to 1, the validation is fully successful. The smaller the index, the less the MCLA domain is contained in the WCA domain.

### 4.4.2. Validation of the security classification against the benchmark

The validation of the security classification carried out by WCA is made by comparison against the security benchmark, according to the general definitions reported above.

### 4.4.3. Validation of security classification against MCLA security classification

Validation of WCA security classification against MCLA security classification consists of checking to what extent WCA is actually more conservative than MCLA. In particular, for a single base case:

a) Consider a **contingency list** consisting of N contingencies.
b) For each contingency *i*:

   i) Set Conservative = 0, NotConservative = 0, Both_Secure = 0, Both_Unsecure = 0

   ii) Run security analysis by WCA
   iii) Run security analysis by MCLA

   iv) If Security_judgement_WCA$_i$ == Unsecure AND Security_judgement_MCLA$_i$ == Secure, then
      increment Conservative by one,
   otherwise
   v) If Security_judgement_WCA$_i$ == Secure AND Security_judgement_MCLA$_i$ == Unsecure, then
      increment NotConservative by one,
   otherwise
   vi) If Security_judgement_WCA$_i$ == Secure AND Security_judgement_MCLA$_i$ == Secure, then
      increment Both_Secure by one,
   otherwise
   vii) If Security_judgement_WCA$_i$ == Secure AND Security_judgement_MCLA$_i$ == Secure, then
      increment Both_Unsecure by one

   c) Compute performance index

   $$\text{Nonconservativeness}_{\text{WCA against MCLA}} = \\ = \text{NotConservative} \;/\; (\text{NotConservative} + \text{Conservative})$$

If the above index or the variable NotConservative are equal to zero, validation is successful. The larger the index or the variable NotConservative, the less WCA is conservative with respect to MCLA.

As far as a comparison of the outputs of WCA and MCLA against the benchmark is concerned, a similar approach can be used to the one described in the following, for FPF validation.

## 4.5. Procedure for validation of MCLA

MCLA output can be evaluated by applying the same classification performance indices of WCA. Moreover, some indices have specifically been defined above for MCLA, namely performance indices dedicated to the individual phenomena concerned with security.

MCLA security assessment relies on the security rules defined offline in WP4. Hence, validation of MCLA is somewhat equivalent to the validation of the rules. Next figure illustrates the procedure. It can be seen that different levels of output are obtained, for each analyzed contingency, namely:

1) sample by sample, per phenomenon;
2) aggregated for all samples, per phenomenon;
3) aggregated for all samples and all phenomena.

In the validation phase, all contingencies must be evaluated, including the ones that would be discarded as harmless (cluster 1) by WCA and the "null contingency".

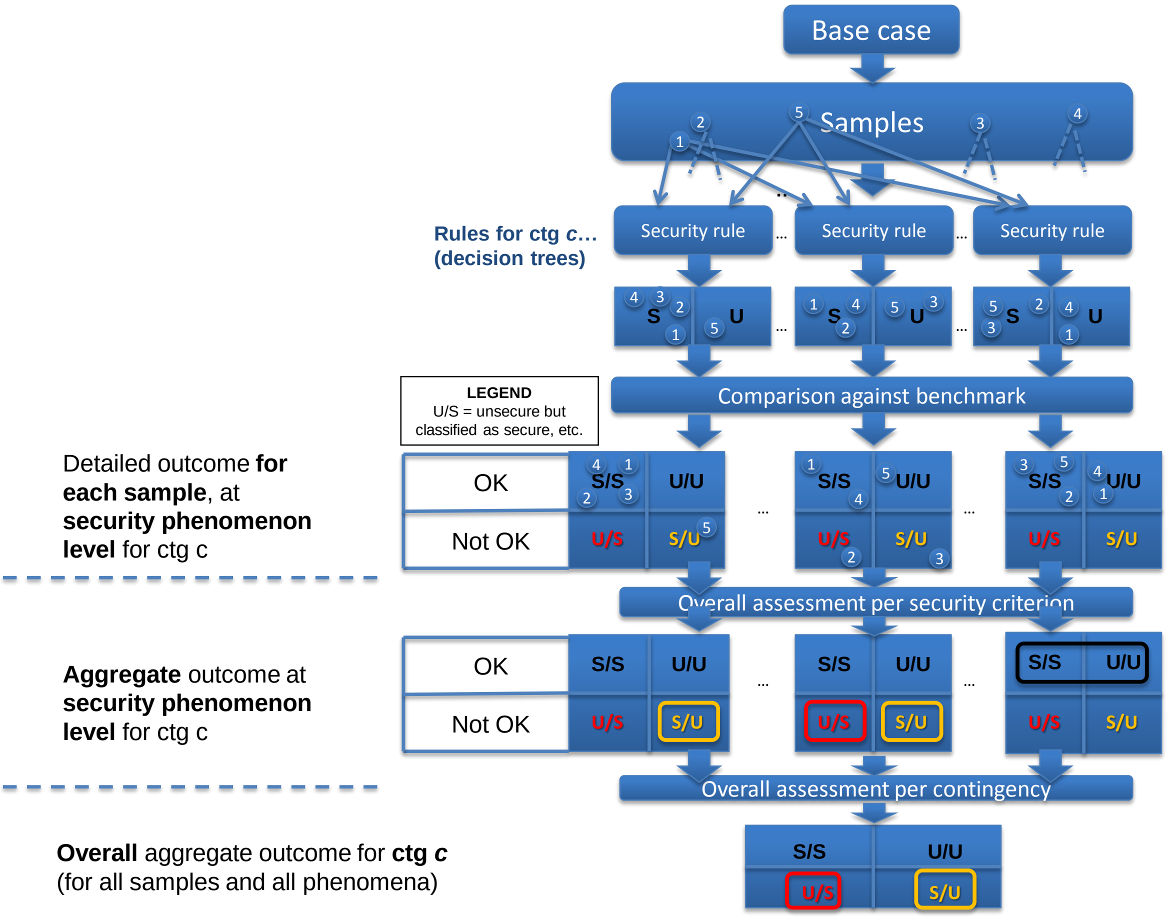


Figure 4 - 3: Performance index computation for MCLA. This process also validates the security rules

## 4.6. Remarks on the quality of the validation set significance

It is worth recalling that the validation cases must be representative of all possible situations that may be encountered in the real operation of the platform. In particular, unsecure cases should be adequately present. This may not be easy to obtain.

Some indices can thus be defined, with the aim to make the validator aware of the significance of the validation set:

- Ratio of unsecure contingencies (calculated from benchmark) wrt the total number of contingencies
  - #unsecure_ctgs / total#ctgs

- Ratio of secure contingencies wrt the total number of contingencies
  - #secure_ctgs / total#ctgs

- Unsecure / secure contingencies ratio
  - #unsecure_ctgs / #secure_ctgs

Moreover, in the output of validation functions it is important to report the absolute figures, not only the above defined indices (which are relative ones): in fact some classes may be very little represented, and it is important to be aware of this.

## 4.7. Validation outputs in the online workflow

In the validation mode, more detailed results are provided, throughout the online process, than would be shown in normal platform operation. These results concern details of the violations, individual security rules output, etc. and they are provided at different steps of the workflow.

Next table summarizes the specific input and outputs in the different operation modes.

**Table 4-1: Detailed reports in validation**

| **Module** | **(states, contingencies)** | | **Outputs stored** | |
|---|---|---|---|---|
| | ***Operator and validation mode*** | ***Validation mode only*** | ***Operator and validation mode*** | ***Validation mode only*** |
| ***Merging*** | Base case files | - | Merged state (base case) | - |
| ***WCA*** | Base case + uncertainty domain<br>All contingencies | - | Cluster for each contingency and details on classification | - |
| ***MCLA*** | Base case + samples<br>Contingencies retained by WCA | All contingencies | • Power flow metrics<br>• pre-contingency steady state violations<br>• post-contingency power flow results (convergence)<br>• post-contingency steady state violations | post-contingency stability judgement by WCA security rules |
| ***Control*** | Set of {State, contingency} retained by MCLA based on static indices | - | List of actions<br>(and possible parameters) | - |
| ***T-D simulation*** | Set of {State, contingency} retained by MCLA based on dynamic indices | All states (i.e. base case + samples), contingencies | • post-contingency stability judgement by security indexes<br>• stabilization and impact analysis metrics | post-contingency stability judgement by security indexes, for unsecure and secure {state, contingency} |

## 4.8. Validation of optimizer

Basic validation of the optimizer would be based on the comparison of the proposed actions against dynamic simulation of the action.

For each contingency **retained** by MCLA:

- For each sampled state:
  - Run optimizer
  - Run T-D simulation with and without corrective actions
  - Check consistency with the outcome of:
    - MCLA (simulations without corrective actions)
    - Optimizer (simulations with corrective actions)
- Identify «real» cluster of the contingency

However, at the moment the validation against dynamic simulation is not planned.

## 4.9. Validation of FPF

Validation of FPF may take place in a two-fold way[6]:

[6] Preliminarily, as the power flow routine used by FPF is different from the ones used in the main workflow, a validation of the power flow routine must be carried out.

1) Check of FPF against results of the filtering process of the main online workflow;
2) Check of FPF against the benchmark.

The individual steps are, for each contingency:

- Run FPF. In case of non-convergence of the power flow, a dynamic instability is assumed.
- Apply the “FPF security rules” and obtain the security judgement by FPF (i.e. secure/unsecure with respect to overloads / voltages).
- Check FPF security assessment results against:
  1) results of the main online workflow;
  2) results of the benchmark.

Performances of the FPF results are evaluated according to the definitions (accuracy, efficiency) provided above.

As far as the comparison with main online workflow is concerned, detailed indices can be defined, focused on the cases that are misclassified by the main workflow and/or by FPF, in order to catch the differences in performances. In particular, the cases of missed and false alarms by main work flow and FPF can be compared and analyzed in detail (see the following figure):

- Missed alarms by FPF, that were correctly classified as unsecure by the Main WF
- Missed alarms by the Main WF, that were correctly classified as unsecure by FPF
- False alarms by FPF, that were correctly classified as secure by the Main WF
- False alarms by the Main WF, that were correctly classified as secure by FPF

Moreover, it is important to investigate the causes of missed alarms as result from both approaches:

- Missed alarms by FPF and by the Main WF

This analysis can help to discover the strengths of each approach. Finally, the comparison of FPF results against the benchmark can also be useful to fine-tune the security thresholds of FPF.

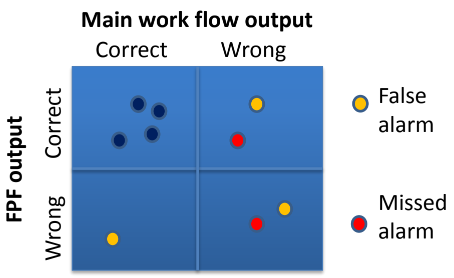


**Figure 4-4 Comparison of contingency analysis results between main online work flow and FPF. Results are classified as “correct” or “wrong” according to comparison against the benchmark.**

# 5. COMMAND LINE USER GUIDE FOR THE OFFLINE PART:

This part is a guide to the beginner user to learn how to use the platform and extract the accurate data for the validation.

## 5.1. How can the platform be accessed?

The platform can be accessed through a linux command window.

An example of command line is:

```
ssh itesla@ns6375578.ovh.net
```

It is also possible to access the platform through a SSH terminal for windows like Putty. Ask the developers or the other validators for the passwords.

## 5.2. What does an offline iTesla command line look like?

A command line syntax is always the same:

```
~/itesla/bin/itools name-of-the-command [--name-of-option1 option1 --name-of-option2 option 2 ...]
```

Not all commands have options. Options can be compulsory or optional.

## 5.3. How can all the existing commands be listed?

Not all existing commands will be listed in this document. The only ones to be listed are the most useful for validation purposes. However, if a special need appears during the validation phase, it is possible to list all the existing commands on the platform to see if one of them can be useful. New commands can be added if necessary.

**List all existing commands**

```
~/itesla/bin/itools
```

This command lists all the existing commands.

**List all options associated with a command**

```
~/itesla/bin/itools name-of-the-command --help
```

This command lists all the options associated with any command. This is a very useful command since the commands and associated options are evolving frequently and the validation handbook can quickly become obsolete.

## 5.4. What are the available command lines?

This part describes a subset list of the available command lines. These are to evolve based on new needs and new developments. All command lines below are examples of real command lines on the French network. Options have to be adapted by the user. Since it is not always easy to adapt French examples on the 7-bus test network, specific examples of command lines on the 7-bus network will be presented in annex.

When an output is displayed by the command, "real" outputs being difficult to read, a simple fictive example will illustrate it.

Please note: in all the outputs listed below, "true" means stable and "false" means unstable. This ambiguous notation is to be changed in the next versions of the tool.

### 5.4.1. WP4 commands

**List workflows**

```
~/itesla/bin/itools list-offline-workflows
```

This command lists all workflows created on the platform. They may be running or not. This is a very useful command to monitor the offline process at any stage.

Example of output: only one workflow called "workflow-0" is created. It is not running.

```
+------------+---------+------+------+
| ID         | Running | Step | Time |
+------------+---------+------+------+
| workflow-0 | false   | IDLE |      |
+------------+---------+------+------+
```

The header meaning is described below.

<u>ID:</u> ID of the workflow.

<u>Running:</u> false or true depending on the state of the workflow.

<u>Step:</u> 4 steps are possible.
- Idle: no calculation is in progress.
- Initialization: preparation of input data in progress (Model parameterization for WP4.1, limits calculation for WP4.2, topology choice etc.).
- Sampling: Process from WP4.1 to WP4.4 in progress.
- Security rules computation: WP4.5 in progress

<u>Time:</u> Remaining time before the end of the current task.

**Create a new workflow**

This command creates a new workflow. The workflow will not start automatically.

Example of command taking into account the default parameters:

```
~/itesla/bin/itools create-offline-workflow --workflow workflow-0
```

Specific command if the user wants to change the default parameters:

```
~/itesla/bin/itools create-offline-workflow --base-case-countries FR --base-case-date 2013-01-15T18:45:00+01:00 --history-interval 2013-01-01T00:00:00+01:00/2013-01-31T23:59:00+01:00 --workflow workflow-0
```

Options :
workflow: name of the workflow to be created
base-case-countries (optional): comma separated list of countries used to define offline workflows geographical perimeter. (ISO code of countries, 2 letters) for several countries, use comma separated list (only FR and BE for now).
base-case-date (optional): date and time of the SN stored in the CIM repository used to define the structure (only) of the network.
history-interval (optional): historical data interval of the SN stored in the Historical DB used for example by the sampling or limits and topologies of the WP4.2 optimizer. Time format corresponds to the ISO 8601 norm. For example, 2013-01-15T18:45:00+01:00 corresponds to January the 15$^{th}$ 2013, 6:45 PM, on time zone UTC+1.

The defaults values for the 3 last options are defined in the “config.xml” file which will be described later in the document.

The output confirms the creation of the workflow. Use “list-offline-workflows” command to visualize the created workflow.

**Create a new simplified workflow**

A simplified version of the offline workflow is available for tests. This version of the workflow only contains the modules from 4.3 to 4.5 which means that the input network situations for the dynamic simulations are not reconstructed (the samples are replaced with the historical data e.g. the SN associated with the historical interval defined when the workflow was created).

To activate this simplification, add the option “—simplified”.

Example:

```
~/itesla/bin/itools create-offline-workflow --workflow workflow-0 –-simplified-workflow
```

**Remove a workflow**

This command removes an existing workflow.

```
~/itesla/bin/itools remove-offline-workflow --workflow workflow-0
```

<u>Options:</u>
workflow: name of the workflow to be removed

No output is displayed. Use “list-offline-workflows” to make sure the workflow was removed.

**Start a workflow**

This command will start the offline process (from WP4.1 to WP4.4) on a specified previously created workflow.

```
~/itesla/bin/itools start-offline-workflow --duration 5 --workflow workflow-0
```

<u>Options:</u>
duration: duration of the simulation in minutes. For now, it includes the initialization which can take a long time so the duration has to be defined carefully.
workflow: name of the workflow to be started.

The base case and historical interval don’t have to be defined again (it was already defined when the workflow was created).

This command starts both the initialization process and WP4.1 to WP4.4 processes. If a simulation was previously generated, the initialization data is already in cache and does not need to be calculated again. When the initialization is launched for the first time, it can take a long time.

When the initialization phase is over, the WP4.1 to WP4.4 phase is processed sets of samples by sets of samples. A set of samples (which number is configured) are generated (WP4.1) and processed all the way to the dynamic simulations phase and indexes generation (WP4.4). When the operation is finished on the first set, the process automatically starts again on the next set of samples and only stops the loops when calculation duration is over. If the defined calculation duration is short, only few samples will be processed (or none is the initialization phase in not complete). Validation DB and Simulation DB are gradually fed (one sample WP4.3 output at a time).

No output is displayed. Use “list-offline-workflows” during the process to make sure the workflow is running.

<u>Particularity of the simplified workflow</u>

When the simplified workflow is used, there is a finite number of network situations to be analyzed (all the SN all the SN associated with the history interval).

Set the “duration” to “-1” to process all the situations. The workflow automatically stops when it is over.

Example:

```
~/itesla/bin/itools start-offline-workflow --duration -1 --workflow workflow-0
```

**Stop a workflow**

This command will stop the offline process on the specified workflow before the duration is over. The command is not executed immediately. The process has to finish on the last sample set before the workflow stops.

```
~/itesla/bin/itools stop-offline-workflow --workflow workflow-0
```

Options:
workflow: name of the workflow to be stopped

No output is displayed. Use “list-offline-workflows” to make sure the workflow has stopped.

**Print security indicators synthesis**

For each studied contingency and for each index, this command prints the number of stable samples and unstable samples contained in the Simulation DB. This is the stability defined by the dynamic simulations. These indicators are not based on the decision trees.

```
~/itesla/bin/itools print-security-indexes-synthesis --workflow workflow-0
```

Options:
workflow: name of the workflow with computed security rules

Example of output for one contingency, 10 samples and 2 indexes:

```
+-------------+---------+---------+
| Contingency | Index_1 | Index_2 |
+-------------+---------+---------+
| N-1_line1   | 3/7     | 9/1     |
+-------------+---------+---------+
```

After a contingency “N-1 line 1”, 7 samples out of 10 are unstable relatively to index 1 and 1 sample out of 10 is unstable relatively to index 2.

**Compute security rules (for all contingency/security index pairs)**

This command calculates the security rules with “DataMaestro” and puts them in the associated DB (Security rules DB).

```
~/itesla/bin/itools compute-security-rules --workflow workflow-0
```

Options:
workflow: name of the workflow to be computed

One security rule is computed per security index-contingency couple from Simulation DB input. For every situation, two security rules are computed: one for the WCSA and one for the MCLA.

No output is displayed. Use “list-offline-workflows” during processing to make sure the rules are being generated.

Many parameters are available to generate the trees. They can be found and changed in config.xml between tags "rulesbuilder". More detailed explanation of the config.xml configuration file can be found in section 7.1.

Here is some information about the parameters:

**missingThreshold**: Since decision trees cannot be trained on missing values, the learning process filters out attributes with too many missing values before starting the actual training.
The "missingThreshold" parameter defines the limit above which an attribute will be entirely discarded from the training set.
E.g. for missingThreshold=0.1 (default value), any attribute that has more than 10% of missing values in the training set will be removed entirely.

**testSetSizeRatio:** Proportion of the data set that belongs to the testing set. Default value = 0.2.

**useCrossValidation:** A true value enables the k-fold Cross validation method. When setting cross-validation to true, the training set will be divided in as many test sets as defined. Then for each test set, the tree will be trained with the complementary set as training set. The global test error will be a mean of all the test sets, and one of the trained trees will be returned.
E.g. for a 20% test set size (default), 5 mutually exclusive test sets will be defined, and 5 trees will be trained, each on the training set that complements the test set.

**alpha:** The alpha value allows to control the size of the tree. The primary target of the alpha value is to avoid to split nodes if there is no statistical significance to split the node objects according to the node test. Alpha = 1 means that the splitting test at each node is always considered statistically significant which leads to a fully developed decision tree. In this case, all leafs are pure but there is a high risk of over-fitting. On the other side, if alpha = 0 the splitting test will always be rejected. In this case the tree will be degenerated to one node. The optimal value of alpha lays in the interval [0;1]. According to D4.4, some literature [1] claims that the optimum alpha value lies between [0.00005; 0.001].
Using a zero alpha value makes no sense and is not supported by the tool. In iTesla toolbox, the possible values for alpha are taken from a fixed list, and can be one of [1, 0.25, 0.1, 0.05, 0.025, 0.01, 0.005, 0.001, 0.0005, 0.00025, 0.0001, 0.00005].

[1] WEHENKEL, PAVELLA M.; "Decision Trees and Transient Stability of Electrical Power Systems", Automatica Vol. 27, N°1, 1991.

**trueThreshold:** Defines how the rules are generated from the decision tree. For the "not pure" leafs (i.e, not totally stable or unstable), defines the threshold for the proportion of "stable" training states for producing a "stable" prediction.
Example: if trueThreshold = 0.95, only the impure leafs with a minimum of 95% of "stable" training states in the leaf provide a "stable" prediction, otherwise the prediction is "unstable". The default value is 0.5. Increasing this value (ex: to 0.95) can reduce the number of misclassified unstable states (i.e., the missed alarms). On the other hand, it can increase the number of misclassified stable states (i.t., the false alarms).

**maxSplits:** This parameter sets the maximum number of splits (test nodes) of the decision tree. The tree grows based on the stop splitting criteria (alpha value) until it doesn't achieve the maximum number of splits. As soon as this maximum value is achieved the decision tree stops the procedure (ex: maxSplits = 1, means that the DT is composed by the root node and two leafs). Considering maxSplit = -1 will disable this stopping criteria.

**Pruning**: A true value enables pruning after learning, which consists in removing subtrees that do not bring extra information when evaluated against a test set. Typically this happens when a certain part of the tree is overfitting, producing good results on the training set, but being irrelevant on the test set.

To have access to information to validate the trees, set debug to “true”. Information about the generation of the decision trees will be available in metrics.csv and in other files that are stored in an output folder. The procedure to find the output folder can be:

1/ View what’s in “~/tmp” folder.
Ex: ls -lrt  ~/tmp
2/ Open the most recent folder « itesla_rules ».
Ex: cd ~/tmp/**itesla_rules_6961151753046427034**

**Compute security rule (for a specific contingency/security index pair for the WCA or MCLA)**

This command calculates a security rule with “DataMaestro” for a specific contingency/security index pair for the WCA or MCLA filtering process. After this computation, the security rule is stored in the associated DB (Security rules DB). Before running the command, when logging in the machine, the following instruction must be executed (to update the java class path):

```
export CLASSPATH=~/itesla/share/java/*:~/itesla/share/java:$CLASSPATH
```

then, the security rule command can be performed as in the following example:

```
~/itesla/bin/itools compute-security-rule --workflow workflow-0 --contingency
N-2_Launa_Taute --index-type TSO_OVERLOAD --attribute-set MONTE_CARLO
```

<u>Options:</u>
workflow: name of the workflow to be computed
contingency: contingency ID
index-type: index ID
attribute-set: WORST_CASE (extracts criteria on active variables only) or MONTE_CARLO (extracts criteria on all variables)

The training parameters are the same already described for the `compute-security-rules` command.

**List computed security rules**

For each studied contingency, this command prints the couples “index-contingency” for which security rules have been computed. If a couple index-contingency is not displayed, it means that the security rule was not generated. It can be a bug during the generation of the security rules.

```
~/itesla/bin/itools list-security-rules --workflow workflow-0
```

Options:
workflow: name of the workflow with computed security rules

Example of output for 2 contingencies and 2 indexes:

```
+----------------+---------------------+
| Contingency ID | Security index type |
+----------------+---------------------+
| N-1_line1      | Index_1             |
+----------------+---------------------+
| N-1_line1      | Index_2             |
+----------------+---------------------+
| N-1_line2      | Index_1             |
+----------------+---------------------+
| N-1_line2      | Index_2             |
+----------------+---------------------+
```

**Print a security rule**

This command will display the security rules associated to one index and one contingency previously calculated with the command “compute-security-rules“ and stored in security rules DB.

```
~/itesla/bin/itools print-security-rule --workflow workflow-0 --attribute-set WORST_CASE --contingency N-2_Waran --index-type TSO_OVERLOAD --purity-threshold 0.95 –-format ASCII_FLAT
```

Options:
workflow: name of the workflow with computed security rules
attribute-set: WORST_CASE (extracts criteria on active variables only) or MONTE_CARLO (extracts criteria on all variables)
contingency: contingency ID
index-type: index ID
[purity-threshold]: optional. Default value is 1.0. This threshold is related to the decision tree generation[7].
[format]: optional. Default value is ASCII_FLAT (output is an expression). Use value ASCII_TREE to have a tree shaped output. Use value GRAPHVIZ_TREE to generate the information to include in a .dot files. This file can be read graphically with open source “GVEdit” software for example.

Example of output of the expression to reach a stable situation:

```
((bus1_P<10)AND(bus2_P<5))OR(bus3_P>8)
```

[7] The purity threshold defines how the rules are generated from the decision tree. A purity threshold of 1 means that leaves nodes which are “not pure” i.e not totally stable or unstable are not taken into account for the generation of the rules. If the purity threshold is lower, it means that a leaf node can be considered as pure if a minimum amount of samples are in the same state.

In reality, expressions can be far longer and complex.

**Visualize decision trees**

This command copies the rules of a selected workflow to the software "DataMaestro" (developed by PEPITE). This command is automatically executed by the "compute-security-rules" command.

```
~/itesla/bin/itools copy-rules --target-rules-db-class eu.itesla_project.histodb.client.impl.RulesDbClientFactoryImpl --workflow workflow-0
```

<u>Options:</u>
workflow: name of the workflow with computed security rules
target-rules-db-class: leave it as in the example

After the command is launched, the decision tree can be read in PEPITE software "DataMaestro".

The GUI for DataMaestro can be accessed at the address: https://ns311581.ovh.net/histodb/rest/iteslasim/. Ask RTE or INESC for the passwords.

To visualize the rules, select the workflow in the list on the left, select tab "Data" and click on "View" for the index wanted. If all the situations are not displayed, change "start=XX" and/or "count=XX" in the field under the tab "Data".

**iteslasim/rules/workflow-workflow-0 >>** Overview Metadata Data Stats

iteslasim/rules/workflow-workflow-0/data? start=0&count=10&headers=true

| algoType | contingencyId | indexType | workflowId | quality | treeSize | criticality | tree |
|---|---|---|---|---|---|---|---|
| WORST_CASE | FSSV_O1__FTILL51__1_ACLS | TSO_synchro_loss | workflow-0 | 1.0 | 1 | 0.0 | View Txt JSON |
| WORST_CASE | FSSV_O1__FTILL51__1_ACLS | TSO_overvoltage | workflow-0 | 1.0 | 1 | 0.0 | View Txt JSON |
| WORST_CASE | FSSV_O1__FTILL51__1_ACLS | TSO_overload | workflow-0 | 0.98173684 | 18 | 0.5754752 | View Txt JSON |
| WORST_CASE | FSSV_O1__FTILL51__1_ACLS | TSO_frequency | workflow-0 | 0.99254566 | 6 | 0.13529631 | View Txt JSON |
| WORST_CASE | FSSV_O1__FTILL51__1_ACLS | SmallSignal | workflow-0 | 1.0 | 1 | 0.0 | View Txt JSON |
| WORST_CASE | FSSV_O1__FTILL51__1_ACLS | Transient | workflow-0 | 1.0 | 1 | 0.0 | View Txt JSON |
| WORST_CASE | FSSV_O1__FTILL51__1_ACLS | TSO_undervoltage | workflow-0 | 1.0 | 1 | 0.0 | View Txt JSON |
| WORST_CASE | FSSV_O1__FTILL51__1_ACLS | OverUnderVoltage | workflow-0 | 0.96807694 | 6 | 0.063846156 | View Txt JSON |
| WORST_CASE | FSSV_O1__FTILL51__1_ACLS | Overload | workflow-0 | 1.0 | 1 | 1.0 | View Txt JSON |
| MONTE_CARLO | FSSV_O1__FTILL51__1_ACLS | TSO_overload | workflow-0 | 0.9988819 | 17 | 0.5754752 | View Txt JSON |

**Figure 5 - 1: DataMaestro snapshot**

In next figure, the colours presented in the decision tree have the following meaning:

- Green – Represent the number of stable training states in each node;
- Orange – Represent the number of unstable training states in each node;
- Red – Indicates the error rate against the test set used at the time of training. A large red bar indicates that the leaf is probably not very meaningful, as it performed poorly against the test set.

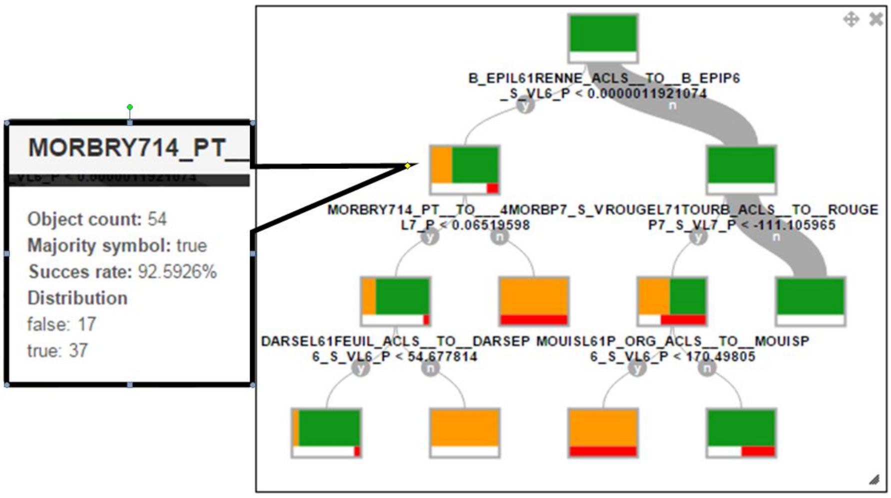


Figure 5 - 2: Decision Tree in DataMaestro

**Export full content of offlinedb to CSV (debugging)**

This command exports in a .csv file data about each sample (active, reactive power, voltage etc. on each and all indexes associated with all contingencies). The information in the file allows the validator to check some key intermediate inputs/outputs. It especially indicates if each sample makes it through every phase from WP4.1 to WP4.4.

```
~/itesla/bin/itools export-security-indexes --workflow workflow-0 --output-file /tmp/offlinedb.csv --add-sample-column --keep-all-samples
```

Options:
workflow: name of the workflow with computed security rules
output-file: /tmp/name_of_output_file.csv
add-sample-column: adds an ID column in the .csv file which allows the user to identify the .xml file in Validation DB associated with any sample in the .csv file.
--keep-all-samples (optional): keep all the samples in the table (even the sample for which all the calculations were not terminated which is very useful for debugging).

The content of this file will be detailed later in the document.

**Export WP4.* module debugging metrics (debugging)**

This command exports in .csv format useful debugging outputs from every WP4.X part.

```
~/itesla/bin/itools export-metrics --workflow workflow-0 --output-file /tmp/metrics.csv
```

Options :
workflow: name of the workflow with computed security rules
output-file: /tmp/name_of_output_file.csv

This file is difficult to understand (even if some of the output information can be understood) but can be useful to developers.

Two columns are worth knowing for a better report of the problems: STARTING_POINT_INITIALIZATION:final_status for WP4.2 and STABILIZATION:error_msg for Eurostag simulations.

Different messages can be found in the STARTING_POINT_INITIALIZATION:final_status column:

If the sample is “OK” after WP4.2:

- “FEASIBLE” means the OPF found a solution within its constraints.
- “FEASIBLE_PC” means the OPF found a solution within its constraints, but with production curtailment.

If the sample is “NOK” after WP4.2:

- “TOPO_NOK_UNFEASIBLE” means that the OPF found that no realistic solution could be found.
- “TOPO_NOK_OPTIMISATIONFAILED” means that the OPF could not find a solution.
- "NF_BALANCE_Q" means the OPF could not adjust reactive power in the network (global balance problem, transits exceeding maximum transmissible power, impossibility to respect voltage constraints…), even without adjusting active power balance nor current constraints.
- "NF_BALANCE_P","NF_BALANCE_LS" mean the OPF could not adjust active power in the network (global balance problem, transits exceeding maximum transmissible power, impossibility to respect voltage constraints…), even without satisfying current constraints.
- "NF_CURRENT_Q" means the OPF could not adjust reactive power in the network (impossibility to respect current constraints), even without adjusting active power balance.
- "NF_CURRENT_P" means the OPF could not adjust active power in the network (impossibility to respect current constraint).
- "FEASIBLE_LS" means the OPF could not find a solution within its constraints without load shedding.
- "FEASIBLE_LS_PC" means the OPF could not find a solution within its constraints without load shedding and production curtailment.
- "NF_MPEC_TD" means the OPF couldn’t determine the values for a discrete variable (phase shift transformers taps).
- "NF_MPEC_UNIT" means the OPF couldn’t determine the values for a discrete variable (generator units).
- "NF_MPEC_SHUNT" means the OPF couldn’t determine the values for a discrete variable (compensation equipments).

The 3 “MPEC” statuses are due to the fact that non linear optimization is not always able to tackle integer constraints. The MPEC method used in WP4.2 is only a heuristic method[8] and may fail.

Many messages can be found in STABILIZATION:error_msg column. The column is particularly helpful to analyze if all problematic samples are discarded with the same error message or not and extract the samples accurately for further analysis on Eurostag.

[8] See D4.2 for more details.

The metrics file also contains the debug information about the generation of the decision trees.

**Export Eurostag data associated with a specific sample**

This command exports all the Eurostag files associated with a given network file (in XML or CIM format) in a defined output folder. This command is useful to analyze a specific sample associated with its network description. In iTesla platform, the IIDM XML files are available in the Validation DB and the CIM files are available in the CIM repository.

```
~/itesla/bin/itools export-eurostag --case-dir ~/france/validation/offline/workflow-0 --case-basename sample-120 --case-format XML --output-dir /tmp/sample_XX_Eurostag
```

Options:
--case-dir: directory where the network file is
--case-basename: network file name without extension
--case-format: file format (CIM1 or XML)
--output-dir: output directory path

The validator has to know that all .xml samples are in the Validation DB. It is not necessary to unzip the files before using the command.

It's better to extract all the Eurostag data in a dedicated folder (sample_XX_Eurostag in the example). The .seq file associated with the stabilization phase is directly generated in the destination folder whereas the .seq files associated with the contingencies are in a specific .zip folder (eurostag-all-scenarios). These .seq files have to be extracted before they are analyzed in Eurostag.

To run the .seq files, it is necessary to remove the "debug" paragraph[9] with a text editor.

**Use Imperial College WP4.1 validation module**

This command generates .png, .fig and .mat files useful for WP4.1 validation.

```
~/itesla/bin/itools wp41-data-comparator --ofile /tmp/WP41_overload/compared_data --set1 [1,4] --set2 [5,8]
```

Options:
ofile: directory and name of the output .png and .fig files
set1: first set of variables to be aggregated and compared (equivalent of "c" input defined in annex 7.12)
set2: second set of variables to be aggregated and compared (equivalent of "d" input defined in annex 7.12)

To understand better this command, please refer to annex 7.12.

[9] The debug part is only supported by the Eurostag version used in the platform and not the commercial one.

### 5.4.2. WP4/WP5 commands

These commands are already part of WP5 but are necessary to WP4 validation.

**Check case security using security rules**

This command will apply the generated security rules to one or more selected network situations.

On the example below, the command applies the security rules on one chosen historical network situation taken from the CIM repository. The command can be run on any network situation.

```
~/itesla/bin/itools check-security --workflow workflow-0 --attribute-set MONTE_CARLO --case-dir ~/caserepo/CIM/FO/2013/01/15 --case-basename 20130115_1830_FO2_FR0 --case-format CIM1
```

Example of output for one file:

```
+-------------+---------+---------+
| Contingency | Index_1 | Index_2 |
+-------------+---------+---------+
| N-1_line1   | OK      | NOK     |
+-------------+---------+---------+
| N-1_line2   | OK      | OK      |
+-------------+---------+---------+
```

On the example below, the command is run on all the XML files (samples) stored in ~/france/validation/offline/workflow-0.

```
~/itesla/bin/itools check-security --case-format XML --case-dir ~/france/validation/offline/workflow-0 --output-csv-file /tmp/out.csv --workflow workflow-0 --attribute-set MONTE_CARLO
```

In this case, the output will be stored in /tmp/out.csv.

<u>Options:</u>
--workflow: workflow on which the security rules were defined
--attribute-set: WORST_CASE (criteria on active power variables only) or MONTE_CARLO (criteria on all variables)
--case-dir: directory where the network file is
--case-basename: network file name without extension (optional: If not included, the command analyzes all the files in the specified directory)
--case-format: file format (CIM1 or XML)
--output-csv-file: specifies the output .csv file where the results are stored (optional: If not included, the outputs are presented in the screen)

Checking security on a list of files can take a long time. To make sure that the calculation goes on even the validator logs out from the server, it can be useful to start the command using “nohup” and “&”.

```
nohup ~/itesla/bin/itools check-security --case-format XML --case-dir ~/france/validation/offline/workflow-0 --output-csv-file /tmp/out.csv --workflow workflow-0 --attribute-set MONTE_CARLO &
```

To monitor the calculation, use for example:

```
tail -f nohup.out
```

**Run a dynamic simulation + security indicators**

This command will run dynamic simulations on a selected network situation (or a list of situations) for all the configured contingencies and re-calculate all the predefined indexes. On the example below, the network situation chosen is taken from the CIM repository but the command can be run on any network situation. This output is to be compared to "check-security" command output to evaluate the quality of the security rules.

On the example below, the command runs a dynamic simulation on one chosen historical network situation taken from the CIM repository. The command can be run on any network situation.

```
~/itesla/bin/itools run-impact-analysis --case-dir ~/caserepo/CIM/FO/2013/01/15
--case-basename 20130115_1830_FO2_FR0 --case-format CIM1
```

Example of output:

```
+-------------+---------+---------+
| Contingency | Index_1 | Index_2 |
+-------------+---------+---------+
| N-1_line1   | OK      | NOK     |
+-------------+---------+---------+
| N-1_line2   | OK      | OK      |
+-------------+---------+---------+
```

On the example below, the command is run on the historical data (FO) from the CIM repository for January 15th 2013.

```
~/itesla/bin/itools run-impact-analysis --case-dir ~/caserepo/CIM/FO/2013/01/15
--output-csv-file /tmp/out.csv --case-format CIM1
```

<u>Options:</u>
--case-dir: directory where the network file is
--case-basename: network file name without extension (optional: If not included, the command analyzes all the files in the specified directory)
--case-format: file format (CIM1 or XML)
--output-csv-file: specifies the output .csv file where the results are stored (optional: If not included, the outputs are presented in the screen)

The output of these commands is supposed to be consistent with the previous "check-security" output if the rules are accurately generated.

It can also be useful to start the command using "nohup" and "&" when using this command on a list of CIM files.

To obtain detailed information about the security indices results, before running the "run-impact-analysis" command, debug must be set to "true" in the eurostag part of the config.xml file. With these settings, the "run-impact-analysis" command will create esg_fs_X.out files (X is the ID number of the contingency) with useful information, like:

- Branches that were classified unsecure by the "TSO_overload" index and violated limits
- Buses that were classified unsecure by the "TSO_Overvoltage" and "TSO_Undervoltage" indexes and violated limits
- Generators that were classified unsecure by the "TSO_Synchroloss", "TSO_Generator_Speed_Automation" and "TSO_Generator_Voltage_Automaton" indexes.

**Validate the rules on a set of data**

This is a specific command to compare easily the output of an index calculated with the simulator configured in the config.xml file (load flow or dynamic simulator) and the same result calculated with security rules on a list of network situations (SN or DACF).
A merging functionality is included in the command so it is possible to define several countries to merge.

The command generates two files in the output folder:

- comparison.csv
- synthesis.csv

"comparisons.csv" contains 2*"number of rules to be tested" + 1 columns. The first one contains the ID of the network situation tested. The second one contains the status of the situation after an index is calculated ("true" = the output of the index is "acceptable", "false" = the output of the index is "not acceptable", "undef" = the simulation crashed). The third column contains the status of the situation after the security rule associated with the couple index/contingency is applied ("true" = the situation is "acceptable" according to the rules, "false" = the situation is "not acceptable" according to the rules, "undef" = the rule could not be applied). The results in columns two and three are replicated for all the situations (index/contingency) to be tested.

"synthesis.csv" contains statistics about the classification errors. 9 kind of situations are calculated (all the combinations between OK_S, NOK_S, OK_R, NOK_R, UNDEF_S and UNDEF_R). "_R" refers to the result of the application of the rule and "_S" refers to the result of the simulation.

Example:

```
~/itesla/bin/itools offline-validation --workflow europetest8 --case-type SN -
-base-case-countries FR,BE,NL,DE --history-interval 2015-01-
01T00:00:00+01:00/2015-01-01T23:59:00+01:00 --output-dir /tmp/
```

The command above applies the rules generated in the workflow "europetest8" to all the SN files of January 1st 2015 for FR, BE, NL and DE. The two output files are created in "/tmp" directory.

The contingencies for which the calculation is made is not specified directly in the command. The command generates the results for all the contingencies listed in the "contingencies.csv" file defined in the config.xml file.

## 5.5. How can the default simulator be changed?

To use Hades instead of Eurostag in the offline workflows you generate, replace the following line in the config.xml file:

Replace

```
<simulatorFactoryClass>eu.itesla_project.eurostag.EurostagFactory</simulatorFactoryClass>
```

With

```
<simulatorFactoryClass>eu.itesla_project.pclfsim.PostContLoadFlowSimFactory</simulatorFactoryClass>
```

Restart the platform:

```
offline-service.sh restart
```

## 5.6. How it is possible to use different configuration files in parallel?

Sometimes, different “itools” commands have to be run on the same server with different configurations.

If there is a need to run a command with a configuration which is different from the main config.xml configuration, use option “--config-name” to define which .xml file has to be used in the command.

For example:

A workflow using the default “config.xml” is running and the validator needs to start an “offline-validation” command with a configuration defined in “config2.xml”.

To use the second configuration on the second command without stopping the workflow which was started previously, use the “--config-name” option:

```
itools offline-validation --config-name config2 …
```

The option can be applied to any command.

## 5.7. How is it possible to deal with an error message?

The logs of the platform can be read to understand which list of actions was just performed by the platform. The .log file path is: ~/itesla/logs/offline.log

To visualize its content in command lines, use for example the command below to print the last 100 lines of the file on the screen.

```
tail -100 ~/itesla/logs/offline.log
```

## 5.8. How can the results of an offline workflow be replicated?

For validation purposes, the results of a workflow can be reproduced by fixing the Matlab seed used for the sampling process. To fix the Matlab seed:

Edit the config file associated with wp4.1 (config.xml).

In config.xml, to set the seed value to 0, add or uncomment the following line between the <samplerwp41> tags: <rngSeed>0</rngSeed>.

## 5.9. How can commands be automated?

For some commands like check-security or run-impact-analysis, one may want to make calculations on several network situations and not just one. These commands will not evolve in themselves to allow it but scripts can be written to launch the commands as many times as necessary.

## 5.1. What is the content of offlinedb.csv?

The file offlinedb.csv is a data file which contains information about the offline process.
Let us consider C contingencies and I indexes tested on a network for which S samples were computed.

There are S+1 lines in the table: one line for each sample and one header line.

The 5 first columns in the table give information about 5 steps in the offline process. The information given for each sample is ‘OK’ or ‘NOK’.

The C*I next columns give the stability state for each couple contingency-index provided by the dynamic simulations.

The next columns give information about the network (active and reactive power on each line, active and reactive power for each load and each production plant, voltage values on each node etc.).

There is no information about the stability provided by the security rules in this file.

Here are excerpts of a generated file:

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | sample | TASK_SAMPLING | TASK_STARTING_POINT_INITIALIZATION | TASK_LOAD_FLOW | TASK_STABILIZATION | TASK_IMPACT_ANALYSIS |
| 2 | 0 | OK | NOK | | | |
| 3 | 1 | OK | NOK | | | |
| 4 | 2 | OK | NOK | | | |
| 5 | 3 | OK | NOK | | | |
| 6 | 4 | OK | OK | OK | OK | OK |
| 7 | 5 | OK | OK | OK | OK | OK |
| 8 | 6 | OK | OK | OK | OK | OK |
| 9 | 7 | OK | OK | OK | OK | OK |
| 10 | 8 | OK | OK | OK | OK | OK |

There was no acceptable starting point for the 4 first samples (WP4.2 is listed as NOK). The next 5 samples went to the whole process.

| | A | G | H | I |
|---|---|---|---|---|
| 1 | sample | SIM_FSSV_O1__FTILL51__1_ACLS__IDX__TSO_overload | SIM_FSSV_O1__FTILL51__1_ACLS__IDX__TSO_undervoltage | SIM_FSSV_O1__FTILL51__1_ACLS__IDX__TSO_overvoltage |
| 29 | 27 | true | false | true |
| 30 | 28 | true | false | true |
| 31 | 29 | true | false | true |
| 32 | 30 | true | false | true |
| 33 | 31 | true | true | true |
| 34 | 32 | true | false | true |
| 35 | 33 | true | false | true |
| 36 | 34 | true | true | true |
| 37 | 35 | true | true | true |

For each couple index - contingency, each sample is listed as stable or unstable.

| | A | P | Q | R | S |
|---|---|---|---|---|---|
| 1 | sample | FP_AND12_L_EC_P | FP_AND12_L_EC_Q | FP_AND12_L_EC_V | FP_AND1_S_VL7_V |
| 2 | 0 | 330.77362 | 32.213528 | 0 | 0 |
| 3 | 1 | 177.65599 | 90.50028 | 0 | 0 |
| 4 | 2 | 190.59311 | 55.434055 | 0 | 0 |
| 5 | 3 | 222.95966 | 89.600464 | 0 | 0 |
| 6 | 4 | 417.64023 | 46.25655 | 405.97443 | 405.97443 |
| 7 | 5 | 325.52426 | 68.07384 | 406.5194 | 406.5194 |
| 8 | 6 | 360.94724 | 27.13122 | 404.12057 | 404.12057 |
| 9 | 7 | 363.51767 | 32.33979 | 406.9099 | 406.9099 |

For each sample, all the values of the variables are available. For the samples for which step WP4.2 was NOK, the saved values are the ones in output of WP4.1 for P and Q, 0 for the other variables.

## 5.2. Is there a graphical user interface?

A graphical user interface was also developed to use the platform. It can be accessed through the address: https://ns6375578.ovh.net/itesla. It is a web application which requires a recent browser.

For now, not every command available through the command line interface can be launched through the graphical interface.

To be completed.

## 5.3. How is it possible to switch between the French network and the 7-bus network in the offline part?

Two different networks coexist in the platform. The French network and a 7-bus test network which will be described later in the document. This is a temporary procedure for the validation phase which is why it is not "clean".

To choose the 7-bus network as a default network, uncomment (remove the # symbols at the head of the lines) the 3 lines starting with "itesla_" on the configuration file "itesla.conf". On the contrary, comment them (add # symbols at the head of the lines) to switch to the French network. The validator doesn't need to understand all the lines in the configuration file "itesla.conf".

To access and change the configuration file, use command line:

```
vi ~/itesla/etc/itesla.conf
```

The content of the configuration file will appear.

The configuration for the 7-bus network is:

```
itesla_cache_dir=/home/itesla/7buses_test/overload/cache
itesla_config_dir=/home/itesla/7buses_test/overload
itesla_config_name=config
mpi_tasks=16
mpi_hosts=localhost
mpirun_options="-mca coll_tuned_use_dynamic_rules 1 -mca
coll_tuned_bcast_algorithm 6 -mca btl_tcp_if_include eth2"
```

The configuration for the French network is:

```
itesla_config_name=config
mpi_tasks=16
mpi_hosts=localhost
mpirun_options="-mca coll_tuned_use_dynamic_rules 1 -mca
coll_tuned_bcast_algorithm 6 -mca btl_tcp_if_include eth2"
```

- itesla_cache_dir is the path to the cache folder. The default value is the path for the French network so it only has to be precised for the 7-bus network.
- itesla_config_dir is the path to the configuration file. The default value is the path for the French network so it only has to be precised for the 7-bus network.
- itesla_config_name is the name of the configuration file. The name used is config.xml but it is possible to test another configuration stored in a file called config2.xml for example and change that name accuratly.
- mpi_tasks is the number of cores (16 per OVH machine).
- mpi_hosts is the server used for the simulation. If you run the workflows on one server only you can write “localhost” or the name of the server explicitly ex: “ns6375582”.
- mpirun_options is to be left as it is.

If any change is made in the configuration file, the platform has to be restarted with command:

```
offline-service.sh restart
```

Then, clean the cache using the script:

```
~/clean_cache.sh
```

The configuration for the 7-bus network was not maintained for a long time (since mid-2015). If you really want to use it, please contact the developers.

## 5.4. How can the version be checked?

The versions of the platform are updated regularly. These updates can sometimes affect the results of tests so it can be useful to know about the version of the code.

To check which version is being used on the platform use command:

```
itools version
```

## 5.5. Examples of lists of commands

The commands listed here apply to the French network. The equivalent list of commands for the 7-bus network is available in annex 7.11.

Here are examples of basic commands to launch tests and analyze results. This example was launched on the "TSO_OVERLOAD" variant.

(1) `itools list-offline-workflows`
(List all the existing workflows)

(2) `itools create-offline-workflow --workflow workflow-0`
(Create offline workflow named workflow-0)

(3) `itools list-offline-workflows`
(Check that the workflow was created)

(4) `itools start-offline-workflow --workflow workflow-0 --duration 720`
(Start the sampling phase on workflow-0 for 12 hours)

(5) `itools list-offline-workflows`
(Check that the workflow is running and wait for the 12 hours to be over)

(6) `itools compute-security-rules --workflow workflow-0`
(Compute the security rules)

(7) `itools print-security-indexes-synthesis --workflow workflow-0`
(Print the security synthesis)

(8) `itools print-security-rule --workflow workflow-0 --attribute-set WORST_CASE --contingency N-2_Launa_Taute --index-type TSO_OVERLOAD --purity-threshold 0.95 --format ASCII_FLAT`
(Prints one security rule for one couple index/contingency)

(9) `itools export-security-indexes --workflow workflow-0 --output-file /tmp/offlinedb.csv --add-sample-column`
(Export validation .csv file in /tmp/offlinedb.csv)

(10) `itools export-metrics --workflow workflow-0 --output-file /tmp/metrics.csv`
(Export debug .csv file in /tmp/metrics.csv)

(11) `itools run-impact-analysis --case-dir ~/france/validation/offline/workflow-0 --case-basename sample-7 --case-format XML`
(Run again eurostag simulations for sample 7)

(12) `~/itesla/bin/itools export-eurostag --case-dir --case-dir ~/france/validation/offline/workflow-0 --case-basename sample-7 --case-format XML --case-format XML --output-dir /tmp/eurostag-7`
(Export eurostag files for sample 7 in /tmp/eurostag-7)

In the list of commands above, the first 6 steps are necessary to obtain raw results from the offline platform and have to be run in this order. The next steps allow the validator to extract useful data from the platform for further results.

Step 7 gives an overview of the repartition of acceptable/unacceptable states for all the couples index/contingency.
Step 8 allows the visualization of the generated rules (to be generated accordingly with the results of step 7). For validation purposes, it can be easier to use DataMaestro interface to visualize the decision trees instead.
Step 9 is very useful to have an overview of the offline process.
Step 10 can help with Eurostag or WP4.2 debugging.

Step 11 Re-runs Eurostag and re-calculates the indicators on a given network situation.
Step 12 Generate all the Eurostag files needed to open them in Eurostag and run the simulations manually.

These last steps are just examples of what can be done with the platform and have to be adapted depending on the needs of the validator.

## 5.6. Specific needs

The paragraphs below detail specific needs which were used by the validators so far. This section is not a priority for beginner users. All the examples below are taken from the French network.

### 5.6.1. Specific commands to have a few details about the indexes:

This lists all the security indexes IDs needed as inputs for the 2nd command.
```
itools simulations-db-list-security-indexes --workflow workflow-0
```

This command saves a .csv file with all the available information for the contingency “N-2_Cergy-Terrier” and the index “smallsignal”.
```
itools simulations-db-print-security-indexes-details --workflow workflow-0 --security-index SIM_N-2_Cergy-Terrier__IDX__SmallSignal > ~/tmp/workflow-0-output.csv
```

For the smallsignal index, the previous command gives information about GMI, SMI and AMI added to the label stable or instable.

This command saves a .csv file with all the available information for the contingency “N-2_DOMLOU_LAUNAY” and the index “TSO_overload”.
```
itools simulations-db-print-security-indexes-details --security-index SIM_N-2_DOMLOU_LAUNAY__IDX__TSO_overload –workflow workflow-0 > SIM_N-2_DOMLOU_LAUNAY__IDX__TSO_overload.csv
```

For the TSO_overload index, the previous command gives information about which lines are overloaded for which sample.

### 5.6.2. Specific commands to run again the KTH indexes in Matlab and analyze them

1/ Check that in config.xml, debug is set to “true” (<debug>true</debug>) in the eurostag part.
2/ Start a « run-impact-analysis » command.
Ex:
```
~/itesla/bin/itools run-impact-analysis --case-dir ~/france/validation/offline/france_demo --case-basename sample-7 --case-format XML
```
3/ View what’s in “~/tmp” folder.

Ex: ls -lrt ~/tmp
4/ Open the most recent folder « itesla_eurostag_impact_analysis ».
Ex: cd ~/tmp/**itesla_eurostag_impact_analysis_7509312567750733345**

In folder in bold characters, you will find the .mat files used as inputs for the KTH indexes. In the example above, for sample “sample-7” and workflow “france_demo”.

For example, “sim_fault_8_smallsignal.mat” corresponds to the smallsignal index for fault 8. The ID number of the contingency is given in order of what is listed in contingencies.csv (in $HOME/france/contingencies.csv) starting from 0.

This folder contains other useful files:

The “slowingvars.csv” files are intermediary steps to determine which lines/transformers were pre-selected as inputs for the small signal index.

Once the list of the generators and nodes on which respectively the speed and voltage vary the most are selected, the list of lines and transformers connected to the selected nodes is established.

The list of selected lines and transformers are labeled “NOT SKIPPED” in the esg_fs_X.out files (X is the ID number of the contingency).

### 5.6.3. Script to change the data between the 4.1 and 4.2 parts

To avoid some crashes in Eurostag due to inconsistencies between static and dynamic data, a script was written (javascript language) to correct data in the offline process.

The script can be found in:

/home/itesla/.itesla/import-post-processor.js

Here are two examples of what to do with the script.

#### 5.6.3.1. Suppress generators

To suppress the « TOTO » generator, add this piece of code:

```
var gen = network.getGenerator("TOTO ");
if (gen != null) {
   gen.remove();
   if (debug)
      print("Generator TOTO removed from network");
}
```

#### 5.6.3.2. Change limits

The voltage limits and the active power limits can be changed. Use the functions defined in the code.

For example, to set the minimum active power to 10 MW for G1 and set the stator voltages between 0.8 and 1.2 pu for G2, write:

changeMinP("G1", 10)
changeVoltageLimitsPu("G2", 0.8, 1.2)

# 6. COMMAND LINE USER GUIDE FOR THE ONLINE PART:

This part is a guide to the beginner user to the online part.

The installation of the online workflow on ns6375578.ovh.net server is in /home/itesla/itesla_wp5 folder.

The itools commands can be run from the bin dir (/home/itesla/itesla_wp5/bin).

Make sure to use "./itools" for running the commands (not just "itools", which would use the offline workflow commands). It is better to use only the online itools commands with this installation (the other commands, i.e. offline related, could have problems due to installation configuration).

Please note that at least with the current installation (it is going to be fixed with the installation on the new servers), before starting the online workflow or the forecast errors analysis that uses mpi it is necessary to **start the online service** (this is not required for the other itools commands):

```
cd /home/itesla/itesla_wp5/bin
./online-service.sh start
```

**To check if the service is running**, use:

```
cd /home/itesla/itesla_wp5/bin
 ./online-service.sh status
```

When the online workflow is over, **stop the service**:

```
cd /home/itesla/itesla_wp5/bin
./online-service.sh stop
```

## 6.1. What does an online iTesla command line look like?

A command line syntax is always the same:

```
~/itesla_wp5/bin/itools name-of-the-command [--name-of-option1 option1 --name-of-option2 option 2 ...]
```

Not all commands have options. Options can be compulsory or optional.

## 6.2. How can all the existing commands be listed?

Not all existing commands will be listed in this document. The only ones to be listed are the most useful for validation purposes. However, if a special need appears during the validation phase, it is possible to list all the existing commands on the platform to see if one of them can be useful.

**List all existing commands**

```
~/itesla_wp5/bin/itools
```

This command lists all the existing commands.

**List all options associated with a command**

```
~/itesla_wp5/bin/itools name-of-the-command --help
```

This command lists all the options associated with any command. This is a very useful command since the commands and associated options are evolving frequently and the validation handbook can quickly become obsolete.

## 6.3. What are the available command lines?

This part describes a subset list of the available command lines. These are to evolve based on new needs and new developments. All command lines below are examples of real command lines on the French network. Options have to be adapted by the user.

### 6.3.1. Run forecast errors analysis for the Monte-Carlo like approach

These commands compute the forecast errors for the Monte-Carlo like approach (MCLA) and some uncertain data (means and standard deviations) for the Fuzzy Power Flow (FPF). The 2 commands run an "offline" forecast error analysis, saving the forecast errors models for later use in the online workflow. The first command uses the local computation platform, while the second one uses the computation platform based on mpi, to distribute the processing.

```
~/itesla_wp5/bin/itools run-forecast-errors-analysis --time-horizon DACF --analysis France_flagPQ1_ir0_80_method_4_nC_20 --base-case-date 2013-01-15T18:30:00.000+01:00 --history-interval 2013-01-01T00:00:00.000+01:00/2013-01-31T23:59:00.000+01:00 --method 4 --outliers 1 --modalityGaussian 0 --flagPQ 1 --percentileHistorical 0.05 --ir 0.80 --nClusters 20 --conditionalSampling 1 --nSamples 100
```

```
~/itesla_wp5/bin/itools run-forecast-errors-analysis-mpi --time-horizon DACF --analysis France_flagPQ1_ir0_80_method_4_nC_20 --base-case-date 2013-01-15T18:30:00.000+01:00 --history-interval 2013-01-01T00:00:00.000+01:00/2013-01-31T23:59:00.000+01:00 --method 4 --outliers 1 --modalityGaussian 0 --flagPQ 1 --percentileHistorical 0.05 --ir 0.80 --nClusters 20 --conditionalSampling 1 --nSamples 100
```

Options:

--time-horizon: time horizon of the forecasts to compare with the corresponding snapshots (SN), to compute the forecast error models (only DACF are supported, so far, by the platform).

--analysis: id of the analysis outputted by this process.

--base-case-date: date of the base case SN used by the analysis. This SN, coming from the CIM repository, is used to get the list of injections (RES and loads) of the network. If the SN is not available, an error is returned.

--history-interval: interval of the historical data used for the analysis (SN and forecasts from historical DB).
--method: method for missing data imputation performed in the pre-processing of real data series (methods properly described in D5.3 – Appendix B). **1** = New method proposed by RSE; **2** = Gaussian conditional sampling; **3** = Gaussian mixture imputation; **4** = Interpolation based method.
--outliers: method of outlier management performed in the pre-processing of real data series. **0** = outliers are included as valid samples, **1** = outliers are excluded.
--modalityGaussian: method for forecast error modeling. **1** = uses a "standard" error model (a normal distribution is assumed for each error variable, being the standard deviation of the error a fixed percentage of the forecast value; variables are sampled as independent; only historical SN are used with this method); **0** = uses historical data (SN and forecasts) with PCA and copula estimation.
--flagPQ: option for reactive power sampling. **1** = P and Q sampled separately; **0** = P sampled and Q computed from P (constant power factor).
--percentileHistorical: quantile of the distribution of historical data related to Q variables, to set realistic limits of Q samples, in case flagPQ = 0 (typical value: 0.05).
--ir: fraction of explained variance for PCA (typical value: 0.8). Used if modalityGaussian = 0.
--nClusters: number of clusters for PCA (typical rule from D4.1: nClusters=sqrt(no. of used historical samples/2)). Used if modalityGaussian = 0.
--conditionalSampling: option of sampling that accounts for dependence of SN for a given forecast configuration **1** = with conditional sampling; **0** = without conditional sampling. Used if modalityGaussian = 0.
-- nSamples: the number of samples to generate, offline. Samples are generated 'offline' for both the conditional and the unconditional cases of the MCLA.

<u>Options only in config.xml file, inside the <forecastErrorsAnalyzer> tag:</u>
- optionSign: option of MCLA sampling. **1** = inversion of the sign of the samples is not allowed for "pure" injection nodes, i.e. for pure load or pure RES generation; **0** = inversion of the sign of the samples is allowed.
- percpuGaussLoad: parameter of the "standard" error model. Percentage of current load forecast, to be used for standard deviation of the "standard" error model (typical value: 0.05).
- percpuGaussRes: parameter of the "standard" error model. Percentage of current RES forecast, to be used for standard deviation of the "standard" error model (typical value: 0.15).
- correlationGauss: parameter of the "standard" error model. Pearson correlation among RES and load variables: **0** = all variables RES and load are uncorrelated; **1** = all variables RES and load are completely correlated; an intermediate value means that RES and LOAD are intermediately correlated, but inside each category (RES and Load) all loads (all RES) are completely correlated (typical value: 0.5).

### 6.3.2. Run forecast errors analysis for the WCA

This command computes the forecast errors for the Worst State Approach (WCA) only. Unlike the forecast errors computation for the Monte-Carlo like approach, if this command is not executed with the accurate parameters before an online workflow is started, the calculation of forecast errors for the WCA will be done automatically in the online workflow. This behavior will impact the duration of the online workflow.

```
~/itesla_wp5/bin/itools analyse-uncertainties --case-basename 20130115_1830_FO2_FR0 --case-dir
/home/itesla/itesla_wp5/france/caserepo/CIM/FO/2013/01/15 --case-format CIM1 --history-interval
2013-01-01T00:00:00+01:00/2013-01-31T23:59:00+01:00 --output-dir /tmp/test_wca
```

<u>Options:</u>
--case-basename: the case base name.

--output-dir: output directory path.
--case-dir: the directory where the case is.
--history-interval: history time interval (example 2013-01-01T00:00:00+01:00/2013-01-31T23:59:00+01:00).
--case-format: the case format [UCTE, XML, CIM1].

All this is stored in the cache file and is included in the online workflow if parameters are consistent.

### 6.3.3. List stored forecast errors analysis results

The command lists the existing forecast errors analysis, stored by the platform. The command presents the date of the analysis and the time horizon used for the analysis.

```
~/itesla_wp5/bin/itools list-forecast-errors-analysis
```

Example of output:

```
+--------------------+---------------------+----------------------+---------------------------+
|         ID         |         Date        | Forecast Errors Data | Forecast Errors Statistics |
+--------------------+---------------------+----------------------+---------------------------+
| 7buses_flagPQ1_ir0_80_v15     | 2015-05-19 17:52:03 | [DACF] | [DACF]                   |
| France_flagPQ1_ir0_80_v15_old | 2015-05-19 18:19:54 | [DACF] | [DACF]                   |
| France_flagPQ1_i r0_80_v15    | 2015-05-29 16:24:49 | [DACF] | [DACF]                   |
| France_flagPQ1_ir0_80_v152  | 2015-06-03 14:56:18 | [DACF]   | [DACF]                   |
+--------------------+---------------------+----------------------+---------------------------+
```

### 6.3.4. Delete a Forecast Errors Analysis

The command deletes a forecast errors analysis stored by the platform and all the stored data related with the analysis.

```
~/itesla_wp5/bin/itools delete-forecast-errors-analysis --time-horizon DACF --analysis France_flagPQ1_ir0_80_method_4
```

Options:
--time-horizon: time horizon (example DACF)
--analysis: ID of the analysis

### 6.3.5. Run online workflow

The command runs an online workflow, providing all the parameters for the specific run. Only one workflow can run at a time.

```
~/itesla_wp5/bin/itools online-workflow-control --start-workflow --base-case 2013-01-15T18:30:00+01:00 --fe-analysis-id France_flagPQ1_ir0_80_v152 --time-horizon DACF --workflow workflow-0 --histodb-interval 2013-01-01T00:00:00+01:00/2013-01-31T23:59:00+01:00 --rules-purity 0.95 --analyse-basecase 1 --states 100 --store-states 1 --validation 1
```

Options:
--base-case: date of the base case forecast under analysis (e.g. 2013-01-15T18:30:00+01:00) coming from the CIM repository.

--basecases-interval: interval for base cases to be considered, coming from the CIM repository. When set, overrides the base-case parameter.
--fe-analysis-id: id of the forecast errors analysis, to be used for the MCLA sampling step.
--time-horizon: time horizon for the MCLA sampling step (this parameter, along with the fe-analysis-id, identifies which error model to use).
--workflow: name of the offline workflow that produced the security rules to be used.
--histodb-interval: interval of the historical data used for the WCA step (e.g. 2013-01-01T00:00:00+01:00/2013-01-31T23:59:00+01:00).
--rules-purity: purity threshold for the security rules[10].
--analyse-basecase: flag (0 or 1) for analyzing the base case with the sampled states. The base case is identified by state 0.
--states: number of states to be sampled by the MCLA. This value must be smaller or equal to the number of samples generated by the forecast error analysis.
--store-states: flag (0 or 1) for the storage of sampled states in the online DB (if =1, the states are stored after MCLA sampling and loadflow).
--validation: flag (0 or 1) for running the online workflow in validation mode. If the validation option is not activated, the workflow is run in the standard mode (i.e. with the filtering process defined by the WP5 methodology, and with only a certain set of results and data stored). In validation mode, the contingencies are not filtered and more results and debug data is stored.

When using a previously computed WP4 workflow, it is important to make sure that the contingencies used to produce the rules are the same than the ones defined in the online part.

#### 6.3.5.1. How can the indexes be filtered?

It is possible to specify which security indexes are used for the analysis in the input parameters of the online workflow. The list is defined by the securityIndexes parameter of the online-default-parameters section (e.g. TSO_OVERLOAD, TSO_SYNCHROLOSS) in the online workflow .XML configuration file. If the value is empty, all the security indexes will be used.

The list of security indexes to use can also be specified as input to the itools command used for starting the online workflow:

```
./itools online-workflow-control --start-workflow --security-indexes
SMALLSIGNAL,TSO_OVERLOAD
```

If no parameter is configured for this option, the value from the config.xml file is used. Use `ALL` as a value to specify the use of all the parameters.

### 6.3.6. List stored online workflows

The command provides a list of the online workflows, stored by the platform, with the date of the run and the parameters used to run them.

---

[10] The purity threshold defines how the rules are generated from the decision tree. For the “not pure” leafs (i.e, not totally safe or unsafe), defines the threshold for the proportion of “safe” training states for producing a “safe” prediction.

```
~/itesla_wp5/bin/itools list-online-workflows
```

Options:
--basecase: date of the considered base case forecast (e.g. 2013-01-15T18:30:00+01:00).
--basecases-interval: interval for the considered base cases.

Example of output (in standard and validation mode):

```
+-------------------+---------------------+--------------------------------------------------------------------------------+
|        ID         |        Date         |                                   Parameters                                   |
+-------------------+---------------------+--------------------------------------------------------------------------------+
| 20150625151301333 | 2015-06-25 15:23:48 | Basecase = 2013-01-01T23:30:00.000+01:00                                       |
|                   |                     | Time Horizon = DACF                                                            |
|                   |                     | FE Analysis Id = France_flagPQ1_ir0_80_v152                                    |
|                   |                     | Offline Workflow Id = 1358271900000-1356994800000-1359673140000                |
|                   |                     | Historical Interval = 2013-01-01T00:00:00.000+01:00/2013-01-31T23:59:00.000+01:00 |
|                   |                     | States = 10                                                                    |
|                   |                     | Rules Purity Threshold = 0.95                                                  |
|                   |                     | Store States = true                                                            |
|                   |                     | Analyse Basecase = true                                                        |
|                   |                     | Validation = false                                                             |
| 20150701100436475 | 2015-07-01 10:16:29 | Basecase = 2013-01-15T18:30:00.000+01:00                                       |
|                   |                     | Time Horizon = DACF                                                            |
|                   |                     | FE Analysis Id = France_flagPQ1_ir0_80_v152                                    |
|                   |                     | Offline Workflow Id = 1358271900000-1356994800000-1359673140000                |
|                   |                     | Historical Interval = 2013-01-01T00:00:00.000+01:00/2013-01-31T23:59:00.000+01:00 |
|                   |                     | States = 10                                                                    |
|                   |                     | Rules Purity Threshold = 0.95                                                  |
|                   |                     | Store States = true                                                            |
|                   |                     | Analyse Basecase = true                                                        |
|                   |                     | Validation = false                                                             |
| 20150702101702793 | 2015-07-02 10:36:19 | Basecase = 2013-01-15T18:30:00.000+01:00                                       |
|                   |                     | Time Horizon = DACF                                                            |
|                   |                     | FE Analysis Id = France_flagPQ1_ir0_80_v152                                    |
|                   |                     | Offline Workflow Id = 1358271900000-1356994800000-1359673140000                |
|                   |                     | Historical Interval = 2013-01-01T00:00:00.000+01:00/2013-01-31T23:59:00.000+01:00 |
|                   |                     | States = 10                                                                    |
|                   |                     | Rules Purity Threshold = 0.95                                                  |
|                   |                     | Store States = true                                                            |
|                   |                     | Analyse Basecase = true                                                        |
|                   |                     | Validation = true                                                              |
+-------------------+---------------------+--------------------------------------------------------------------------------+
```

### 6.3.7. Print the processing status of the steps of an online workflow

The command prints the processing status (OK, FAILED) of the different steps for the analyzed states of an online workflow.

```
~/itesla_wp5/bin/itools print-online-workflow-processing-status --workflow 20150508131613965
```

Options:
--workflow: id of the online workflow.
--csv: print in CSV format, instead of tabular format.

Example of output (in standard and validation mode):

| State | Montecarlo Sampling | Loadflow | Security Rules | Optimizer | T-D Simulation | Detail |
|---|---|---|---|---|---|---|
| 0 | OK | OK | OK | OK | FAILED | time-domain simulation failed (stabilization): metrics = {badExitCode=1, initialValueErrors=[{"machine":"SAUS26G1","macrobloc":"EDFTURHC","variable":"","blocdesort":"106","macrobloc":"EDFTURHC","limiteinf":"0.00000E+00","valeur":"0.11142E+01","limitesup":"0.11000E+01"}], error={"mod":"0227","num":"0512","msg":"CERTAINES VARIABLES SOUMISES A LIMITATIONS SONT EN DEHORS DE LEURS BORNES "}, lf_diverge=false, lf_iterations=9, stab_rich_status=FAILED} |
| 1 | OK | OK | OK | OK | FAILED | time-domain simulation failed (stabilization): metrics = {steadyStateErrors=[{"machine":"MANOS6GR","macrobloc":"RT_HYDRH","variable":"D(SREGUL )/DT","blocdesort":"1","valeurPu":19.0356},{"machine":"REVI57G2","macrobloc":"RT_HYDRH","variable":"D(SREGUL )/DT","blocdesort":"1","valeurPu":21.2942},{"machine":"REVI57GR","macrobloc":"RT_HYDRH","variable":"D(SREGUL )/DT","blocdesort":"1","valeurPu":25.5138},{"machine":"REVI57G0","macrobloc":"RT_HYDRH","variable":"D(SREGUL )/DT","blocdesort":"1","valeurPu":27.4429},{"machine":"HERMI6G2","macrobloc":"RT_HYDRH","variable":"D(SREGUL )/DT","blocdesort":"1","valeurPu":297.2695},{"machine":"COGE36A0","macrobloc":"AMFAREGU","variable":"","blocdesort":"21","valeurPu":904.2578}], lf_diverge=false, lf_iterations=7, stab_rich_status=COMPLETED_BUT_NOT_TO_STEADY_STATE} |
| 2 | OK | OK | OK | OK | OK | - |
| 3 | OK | OK | OK | OK | OK | - |
| 4 | OK | OK | OK | OK | OK | - |

### 6.3.8. Print the Online Workflow Failures and Violations Summary Table

This command allows printing a summary table, listing all the failures and violations of online workflows:

```
./itools print-online-workflow-summary  --workflow 20151111161513748 --output-file
~/itesla_wp5/teste.csv
```

<u>Options:</u>
--workflow: the workflow id
--workflows: the workflow ids, separated by ,
--basecase: date of the considered base case forecast (e.g. 2013-01-15T18:30:00+01:00).
--basecases-interval: interval for the considered base cases.
--output-file: the output .csv file

The command can be run on a specific workflow, on a list of workflows, on all the workflows run on a specific base case (base case date), or on all the workflows run on base cases included in an interval of dates.

The command print in the output file (CSV format), the following information:
`WorkflowId, Basecase, Contingency, State, FailureStep, FailureDescription, ViolationType, Violation, ViolationStep, Equipment, Value, Limit.`

ViolationType can be STEADY_STATE, WCA_RULE, MCLA_RULE, SECURITY_INDEX. Violation depends on the ViolationType: for STEADY_STATE can be CURRENT, LOW_VOLTAGE and HIGH_VOLTAGE, for the other ones is the name of the index (e.g. TSO_overload). Equipment, Value and Limit are only populated for STEADY_STATE violations.

Example of output in standard mode:

| WorkflowId | Basecase | Contingency | State | Failure Step | Failure Description | ViolationType | Violation | ViolationStep | Equipment | Value | Limit |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 20130115_1830_20151204133654861 | 20130115_1830_FO2_FR0 | Empty-Contingency | 0 | | | STEADY_STATE | LOW_VOLTAGE | LOAD_FLOW | M_PONP7_S_VL7 | 387.25363 | 391 |
| 20130115_1830_20151204133654861 | 20130115_1830_FO2_FR0 | N-1_BRAUDL71PREGU | 0 | | | STEADY_STATE | LOW_VOLTAGE | POSTCONTINGENCY_LOAD_FLOW | M_PONP7_S_VL7 | 387.3094 | 391 |
| 20130115_1830_20151204133654861 | 20130115_1830_FO2_FR0 | N-1_BRAUDL71PREGU | 0 | | | STEADY_STATE | LOW_VOLTAGE | POSTCONTINGENCY_LOAD_FLOW | PONTEP7_S_VL7 | 387.30905 | 391 |
| 20130115_1830_20151204133654861 | 20130115_1830_FO2_FR0 | N-1_Tavel-Tama | 0 | | | STEADY_STATE | LOW_VOLTAGE | POSTCONTINGENCY_LOAD_FLOW | M_PONP7_S_VL7 | 386.78848 | 391 |
| 20130115_1830_20151204133654861 | 20130115_1830_FO2_FR0 | N-1_Tavel-Tama | 0 | | | STEADY_STATE | LOW_VOLTAGE | POSTCONTINGENCY_LOAD_FLOW | PONTEP7_S_VL7 | 386.78818 | 391 |
| 20130115_1830_20151204133654861 | 20130115_1830_FO2_FR0 | N-2_Launa_Taute | 0 | | | MCLA_RULE | TSO_overload | MONTE_CARLO_LIKE_APPROACH | | | |
| 20130115_1830_20151204133654861 | 20130115_1830_FO2_FR0 | N-2_Launa_Taute | 0 | | | STEADY_STATE | CURRENT | POSTCONTINGENCY_LOAD_FLOW | DRONNL61FLERS_ACLS | 1823.6686 | 1375 |
| 20130115_1830_20151204133654861 | 20130115_1830_FO2_FR0 | N-2_Launa_Taute | 0 | | | STEADY_STATE | CURRENT | POSTCONTINGENCY_LOAD_FLOW | FLERSL61LAUNA_ACLS | 948.6047 | 781 |
| … | … | … | … | … | … | … | … | … | … | … | … |

Example of output in validation mode:

| WorkflowId | Basecase | Contingency | State | FailureStep | FailureDescription | ViolationType | Violation | ViolationStep | Equipment | Value | Limit |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 20130115_1830_2015120413164694 | 20130115_1830_FO2_FR0 | Empty-Contingency | 0 | TIME_DOMAIN_SIM | time-domain simulation failed (stabilization): metrics = {badExitCode=1, initialValueErrors=[{"machine":"SAUS26G1","macrobloc":"EDFTURHC","variable":"","blocdesort":"106","macrobloc":"EDFTURHC","limiteinf":"0.00000E+00","valeur":"0.11142E+01","limitesup":"0.11000E+01"}], error={"mod":"0227","num":"0512","msg":"CERTAINES VARIABLES SOUMISES A LIMITATIONS SONT EN DEHORS DE LEURS BORNES "}, lf_diverge=false, lf_iterations=9, stab_rich_status=FAILED} | | | | | | |
| 20130115_1830_2015120413164694 | 20130115_1830_FO2_FR0 | Empty-Contingency | 0 | | | STEADY_STATE | LOW_VOLTAGE | LOAD_FLOW | M_PONP7_S_VL7 | 387.25363 | 391 |
| 20130115_1830_2015120413164694 | 20130115_1830_FO2_FR0 | N-1_BRAUDL71PREGU | 0 | | | STEADY_STATE | LOW_VOLTAGE | POSTCONTINGENCY_LOAD_FLOW | M_PONP7_S_VL7 | 387.3094 | 391 |
| 20130115_1830_2015120413164694 | 20130115_1830_FO2_FR0 | N-1_BRAUDL71PREGU | 0 | | | STEADY_STATE | LOW_VOLTAGE | POSTCONTINGENCY_LOAD_FLOW | PONTEP7_S_VL7 | 387.30905 | 391 |
| 20130115_1830_2015120413164694 | 20130115_1830_FO2_FR0 | N-1_Tavel-Tama | 0 | | | STEADY_STATE | LOW_VOLTAGE | POSTCONTINGENCY_LOAD_FLOW | M_PONP7_S_VL7 | 386.78848 | 391 |
| 20130115_1830_2015120413164694 | 20130115_1830_FO2_FR0 | N-1_Tavel-Tama | 0 | | | STEADY_STATE | LOW_VOLTAGE | POSTCONTINGENCY_LOAD_FLOW | PONTEP7_S_VL7 | 386.78818 | 391 |
| 20130115_1830_2015120413164694 | 20130115_1830_FO2_FR0 | N-2_Launa_Taute | 0 | | | WCA_RULE | TSO_overload | MONTE_CARLO_LIKE_APPROACH | | | |
| 20130115_1830_2015120413164694 | 20130115_1830_FO2_FR0 | N-2_Launa_Taute | 0 | | | MCLA_RULE | TSO_overload | MONTE_CARLO_LIKE_APPROACH | | | |
| 20130115_1830_2015120413164694 | 20130115_1830_FO2_FR0 | N-2_Launa_Taute | 0 | | | STEADY_STATE | CURRENT | POSTCONTINGENCY_LOAD_FLOW | DRONNL61FLERS_ACLS | 1823.6686 | 1375 |
| … | … | … | … | … | … | … | … | … | … | … | … |

### 6.3.9. Print the Online Workflow Performances

This command computes and prints the performances of the online workflows if the validation mode has been activated:

```
./itools print-online-workflow-performances  --workflow 20151111161513748 --csv-file ~/itesla_wp5/teste.csv
```

Options:

--workflow: the workflow id.

--workflows: the workflow ids, separated by ,

--basecase: date of the considered base case forecast (e.g. 2013-01-15T18:30:00+01:00).
--basecases-interval: the base cases interval.
--csv-file: the ouptput CSV file.

The command can be run on a specific workflow, on a list of workflows, on all the workflows run on a specific base case (base case date), or on all the workflows run on base cases included in an interval of dates.

The command prints, in the csv file, the following information:
`workflow_id, basecase, secure_contingencies, unsecure_contingencies, unsecure_contingencies_ratio, secure_contingencies_ratio, unsecure_secure_contingencies_ratio, wca_missed_alarms, wca_missed_alarms_lst, wca_false_alarms, wca_false_alarms_lst, wca_accuracy, wca_efficiency, mcla_missed_alarms, mcla_missed_alarms_lst, mcla_false_alarms, mcla_false_alarms_lst, mcla_accuracy, mcla_efficiency, wf_missed_alarms, wf_missed_alarms_lst, wf_false_alarms, wf_false_alarms_lst, wf_accuracy,wf_efficiency`

The explanations of these fields are presented in section 4.

Example of output (in validation mode):

| | |
|---|---|
| workflow_id | 20130115_1830_20151204131644694 |
| basecase | 20130115_1830_FO2_FR0 |
| secure_contingencies | 1 |
| unsecure_contingencies | 2 |
| unsecure_contingencies_ratio | 0.6666667 |
| secure_contingencies_ratio | 0.33333334 |
| unsecure_secure_contingencies_ratio | 2 |
| wca_missed_alarms | 0 |
| wca_missed_alarms_lst | [] |
| wca_false_alarms | 1 |
| wca_false_alarms_lst | [N-1_Tavel-Tama] |
| wca_accuracy | 100 |
| wca_efficiency | 0 |
| mcla_missed_alarms | 0 |
| mcla_missed_alarms_lst | [] |
| mcla_false_alarms | 0 |
| mcla_false_alarms_lst | [] |
| mcla_accuracy | 100 |
| mcla_efficiency | 100 |
| wf_missed_alarms | 0 |
| wf_missed_alarms_lst | [] |
| wf_false_alarms | 0 |
| wf_false_alarms_lst | [] |
| wf_accuracy | 100 |
| wf_efficiency | 100 |

### 6.3.10. Print the results of the WCA step of an online workflow

The command prints the stored results of the WCA step of an online workflow: cluster number per contingency. The clusters are:
1 – secure;
2 – requires only corrective actions;
3 – requires corrective and preventive actions;
4 – requires load shedding.

This command also provides a cause for the assignment of a contingency into a cluster. This cause is printed along with the WCA results.

```
~/itesla_wp5/bin/itools print-online-workflow-wca-results --workflow 20150625151301333
```

Options:
--workflow: id of the online workflow.
--csv: print in CSV format, instead of tabular format.

Example of output (in standard and validation mode):

```
+------------------+-----------+-----------+-----------+-----------+-----------+---------------------------------------------------+
|    Contingency   | Cluster 1 | Cluster 2 | Cluster 3 | Cluster 4 | Undefined |                       Cause                       |
+------------------+-----------+-----------+-----------+-----------+-----------+---------------------------------------------------+
| N-1_BRAUDL71PREGU |     -     |     -     |     -     |     X     |     -     |                                                   |
| N-2_Launa_Taute  |     -     |     -     |     -     |     X     |     -     | [Rule TSO_OVERLOAD (WORST_CASE) is always unsecure] |
| N-1_Tavel-Tama   |     -     |     -     |     -     |     X     |     -     |                                                   |
+------------------+-----------+-----------+-----------+-----------+-----------+---------------------------------------------------+
```

### 6.3.11. Print the results of the security rules application for an online workflow

The command prints the stored results of the security rules application (from MCLA or WCA) of an online workflow: security status per contingency, per state and per phenomena. The status of the [contingency,state] pair (SAFE, SAFE WITH CORRECTIVE ACTIONS, UNSAFE) is computed according to the results of the security status of the different phenomena, i.e. Unsafe for dynamic phenomena (e.g. small signal stability) leads to UNSAFE status, while Unsafe for static phenomena (e.g. overload) leads to SAFE WITH CORRECTIVE ACTIONS.

However, only if the validation parameter of the online workflow configuration is set to true, the online workflow stores also the results of the application of the WCA security rules to all samples (including the base case, state 0).

```
~/itesla_wp5/bin/itools print-online-workflow-rules-results --workflow 20150625151301333
```

Options:
--workflow: id of the online workflow.
--csv: export in CSV format, instead of tabular format (in tabular format if not specified).
--wca: get results of WCA rules (for Monte Carlo if not specified).

If adding the `--wca` parameter, the command prints the results of the WCA rules, instead of the MCLA rules.

Example of output (in standard and validation mode):

| Contingency | State | Status | SmallSignal | TSO_overload | TSO_overvoltage | TSO_undervoltage | TSO_frequency |
|---|---|---|---|---|---|---|---|
| N-2_Launa_Taute | 0 | UNSAFE | Unsafe | Safe | Unsafe | Unsafe | Safe |
| N-2_Launa_Taute | 1 | UNSAFE | Unsafe | Safe | Unsafe | Unsafe | Safe |
| N-2_Launa_Taute | 2 | UNSAFE | Unsafe | Safe | Unsafe | Unsafe | Safe |
| N-2_Launa_Taute | 3 | UNSAFE | Unsafe | Safe | Unsafe | Unsafe | Safe |
| N-2_Launa_Taute | 4 | UNSAFE | Unsafe | Safe | Unsafe | Unsafe | Safe |
| N-2_Avoi_Distr | 0 | UNSAFE | Unsafe | Safe | Safe | Unsafe | Safe |
| N-2_Avoi_Distr | 1 | UNSAFE | Unsafe | Safe | Safe | Unsafe | Safe |
| N-2_Avoi_Distr | 2 | UNSAFE | Unsafe | Safe | Unsafe | Unsafe | Safe |
| N-2_Avoi_Distr | 3 | UNSAFE | Unsafe | Safe | Unsafe | Unsafe | Safe |
| N-2_Avoi_Distr | 4 | UNSAFE | Unsafe | Safe | Safe | Unsafe | Safe |
| N-2_Waran | 0 | SAFE_WITH_CORRECTIVE_ACTIONS | Safe | Safe | Unsafe | Unsafe | Safe |
| N-2_Waran | 1 | SAFE_WITH_CORRECTIVE_ACTIONS | Safe | Unsafe | Unsafe | Unsafe | Safe |
| N-2_Waran | 2 | SAFE_WITH_CORRECTIVE_ACTIONS | Safe | Safe | Unsafe | Unsafe | Safe |
| N-2_Waran | 3 | SAFE | Safe | Safe | Safe | Safe | Safe |
| N-2_Waran | 4 | SAFE_WITH_CORRECTIVE_ACTIONS | Safe | Safe | Unsafe | Safe | Safe |
| N-2_Rouge_Barna | 0 | UNSAFE | Unsafe | Safe | Unsafe | Unsafe | Safe |
| N-2_Rouge_Barna | 1 | UNSAFE | Unsafe | Safe | Unsafe | Unsafe | Safe |
| N-2_Rouge_Barna | 2 | UNSAFE | Unsafe | Safe | Safe | Unsafe | Safe |
| N-2_Rouge_Barna | 3 | UNSAFE | Unsafe | Safe | Safe | Unsafe | Safe |
| N-2_Rouge_Barna | 4 | UNSAFE | Unsafe | Safe | Safe | Unsafe | Safe |
| N-1_Tavel-Tama | 0 | SAFE_WITH_CORRECTIVE_ACTIONS | Safe | Safe | Unsafe | Unsafe | Safe |
| N-1_Tavel-Tama | 1 | SAFE_WITH_CORRECTIVE_ACTIONS | Safe | Safe | Unsafe | Unsafe | Safe |
| N-1_Tavel-Tama | 2 | SAFE_WITH_CORRECTIVE_ACTIONS | Safe | Safe | Safe | Unsafe | Safe |
| N-1_Tavel-Tama | 3 | SAFE_WITH_CORRECTIVE_ACTIONS | Safe | Safe | Unsafe | Unsafe | Safe |
| N-1_Tavel-Tama | 4 | UNSAFE | Unsafe | Safe | Unsafe | Unsafe | Safe |

### 6.3.12. Print the results of the corrective control optimizer of an online workflow

The command prints the stored results of the corrective control optimizer of an online workflow: list of corrective actions per contingency and state. The corrective actions are read from an XML file, referred in the config.xml.

```
~/itesla_wp5/bin/itools print-online-workflow-optimizer-results --workflow 20150625151301333
```

<u>Options:</u>
--workflow: id of the online workflow.
--csv: print in CSV format, instead of tabular format.

Example of output (in standard and validation mode):

| Contingency | State | Actions |
|---|---|---|
| N-2_Launa_Taute | 0 | LAUNAL72TAUTE_ACLS_opening[boolean_action_param=true,string_action_param='value',float_action_param=0.0f];LAUNAL71TAUTE_ACLS_opening[boolean_action_param=true,string_action_param='value',float_action_param=0.0f] |
| N-2_Launa_Taute | 1 | LAUNAL72TAUTE_ACLS_opening[boolean_action_param=true,string_action_param='value',float_action_param=0.0f];LAUNAL71TAUTE_ACLS_opening[boolean_action_param=true,string_action_param='value',float_action_param=0.0f] |
| N-2_Launa_Taute | 2 | LAUNAL72TAUTE_ACLS_opening[boolean_action_param=true,string_action_param='value',float_action_param=0.0f];LAUNAL71TAUTE_ACLS_opening[boolean_action_param=true,string_action_param='value',float_action_param=0.0f] |
| N-2_Launa_Taute | 3 | LAUNAL72TAUTE_ACLS_opening[boolean_action_param=true,string_action_param='value',float_action_param=0.0f];LAUNAL71TAUTE_ACLS_opening[boolean_action_param=true,string_action_param='value',float_action_param=0.0f] |
| N-2_Launa_Taute | 4 | LAUNAL72TAUTE_ACLS_opening[boolean_action_param=true,string_action_param='value',float_action_param=0.0f];LAUNAL71TAUTE_ACLS_opening[boolean_action_param=true,string_action_param='value',float_action_param=0.0f] |
| N-1_BRAUDL71PREGU | 2 | BRAUDL71PREGU_ACLS_opening[boolean_action_param=true,string_action_param='value',float_action_param=0.0f] |
| N-1_BRAUDL71PREGU | 3 | BRAUDL71PREGU_ACLS_opening[boolean_action_param=true,string_action_param='value',float_action_param=0.0f] |
| N-1_BRAUDL71PREGU | 4 | BRAUDL71PREGU_ACLS_opening[boolean_action_param=true,string_action_param='value',float_action_param=0.0f] |

The table lists just the states analysed by the optimizer, according to MCLA output. The states, according to the optimizer, should become stable if the actions are applied. If the optimizer did not find actions for a state and a contingency, the [state, contingency] pair is analysed using the TD simulation and it will not appear in this table.

Here are the different exit status of the optimizer which can occur:

| **Error code** | **Message to the dispatcher** | **Diagnostic** |
|---|---|---|
| MANUAL_CORRECTIVE_ACTION_FOUND | Specific corrective actions have to be implemented by the dispatcher | SUCCESS<br>➔ TAKE ACTION |
| AUTOMATIC_CORRECTIVE_ACTION_FOUND | Corrective actions are automatically implemented | SUCCESS<br>➔ NOTHING TO DO |
| NO_CORRECTIVE_ACTION_FOUND | The proposed specific corrective actions are not able to relieve the constraints violation | FAILURE<br>➔ ENRICH THE KNOWLEDGE DATABASE |
| NO_SUPPORTED_CORRECTIVE_ACTION_AVAILABLE_IN_THE_DATABASE | No specific corrective action found in the catalogue for that contingency and the violated constraints | FAILURE<br>➔ ENRICH THE KNOWLEDGE DATABASE |
| NO_CONSTRAINT_VIOLATED | No post-contingency constraint violation found | FAILURE<br>➔ CONTACT SUPPORT |
| OPTIMIZER_EXECUTION_ERROR | Optimization module error (know issue) | FAILURE<br>➔ CONTACT SUPPORT |
| OPTIMIZER_INTERNAL_ERROR | Optimization module error (crash) | FAILURE<br>➔ CONTACT SUPPORT |

### 6.3.13. Print the results of the T-D simulation step of an online workflow

The command prints the stored results of the time-domain simulation step of an online workflow: security status per contingency, per state and per phenomena.

```
~/itesla_wp5/bin/itools print-online-workflow-simulation-results --workflow 20150625151301333
```

<u>Options:</u>
--workflow: id of the online workflow.
--csv: print in CSV format, instead of tabular format.

Example of output (in standard and validation mode):

| Contingency | State | Overload | OverUnderVoltage | SmallSignal | Transient | TSO_frequency |
|---|---|---|---|---|---|---|
| N-2_Launa_Taute | 0 | Unsafe | Safe | Safe | Safe | Safe |
| N-2_Launa_Taute | 1 | Unsafe | Safe | Safe | Safe | Safe |
| N-2_Launa_Taute | 2 | Unsafe | Safe | Safe | Safe | Safe |
| N-2_Launa_Taute | 5 | Unsafe | Safe | Unsafe | Safe | Safe |
| N-2_Launa_Taute | 6 | Unsafe | Safe | Safe | Safe | Safe |
| N-2_Launa_Taute | 7 | Unsafe | Safe | Safe | Safe | Safe |
| N-2_Launa_Taute | 8 | Unsafe | Safe | Safe | Safe | Safe |
| N-2_Launa_Taute | 9 | Unsafe | Safe | Safe | Safe | Safe |
| N-2_Avoi_Distr | 0 | Unsafe | Safe | Safe | Safe | Safe |
| N-2_Avoi_Distr | 1 | Unsafe | Safe | Safe | Safe | Safe |
| N-2_Avoi_Distr | 2 | Unsafe | Unsafe | Safe | Safe | Safe |
| N-2_Avoi_Distr | 5 | Unsafe | Safe | Safe | Safe | Safe |
| N-2_Avoi_Distr | 6 | Unsafe | Safe | Safe | Safe | Safe |
| N-2_Avoi_Distr | 7 | Unsafe | Safe | Safe | Safe | Safe |
| N-2_Avoi_Distr | 8 | Unsafe | Safe | Safe | Safe | Safe |
| N-2_Avoi_Distr | 9 | Unsafe | Safe | Safe | Safe | Safe |
| N-2_Rouge_Barna | 0 | Unsafe | Safe | Safe | Safe | Safe |
| N-2_Rouge_Barna | 1 | Unsafe | Safe | Safe | Safe | Safe |
| N-2_Rouge_Barna | 2 | Unsafe | Safe | Safe | Safe | Safe |
| N-2_Rouge_Barna | 5 | Unsafe | Safe | Safe | Safe | Safe |
| N-2_Rouge_Barna | 6 | Unsafe | Safe | Safe | Safe | Safe |
| N-2_Rouge_Barna | 7 | Unsafe | Safe | Safe | Safe | Safe |
| N-2_Rouge_Barna | 8 | Unsafe | Safe | Safe | Safe | Safe |
| N-2_Rouge_Barna | 9 | Unsafe | Safe | Safe | Safe | Safe |
| N-1_Tavel-Tama | 5 | Unsafe | Safe | Safe | Safe | Safe |
| N-1_Tavel-Tama | 6 | Unsafe | Safe | Safe | Safe | Safe |

If the workflow is not run in validation mode, the time-domain simulation is run only for the [state, contingency] pairs that present dynamic issues, according to MCLA output (so, they were not given as input to IPSO), or for the pairs IPSO was not able to find corrective actions.

### 6.3.14. Delete an online workflow

The command deletes an online workflow stored by the platform, along with all the workflow stored data (results and states).

```
~/itesla_wp5/bin/itools delete-online-workflow --workflow 20150625151301333
```

<u>Options:</u>
--workflow: id of the online workflow.

### 6.3.15. List stored states of an online workflow

The command prints a list of stored sampled states of an online workflow. The sampled states of an online workflow are stored, after Monte Carlo sampling and load flow, if the proper online workflow parameter is set to true (see online workflow parameters).

```
~/itesla_wp5/bin/itools list-online-workflow-states --workflow 20150625151301333
```

<u>Options:</u>
--workflow: id of the online workflow.

Example of output (in standard and validation mode):

```
Stored States = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

### 6.3.16. Delete online workflow states

The command deletes the stored states of an online workflow.

```
~/itesla_wp5/bin/itools delete-online-workflow-states --workflow 20150625151301333
```

Options:
--workflow: id of the online workflow.

### 6.3.17. Export a predefined online workflow state

The command export, in CSV format, the network data (load flow results) of the stored states of an online workflow. The command writes data in 6 files: buses.csv, lines.csv, tfos2w.csv, tfos3w.csv, generators.csv and loads.csv, containing the data of the different equipment's of the network.

```
~/itesla_wp5/bin/itools export-online-workflow-state --workflow 20150625151301333 --state 0 --
folder state-0
```

Options:
--workflow: id of the online workflow.
--state: id of the online workflow state.
--folder: folder where to store the CSV files. It must exist. If not provided, or if it does not exists, the files are stored in the current folder.

### 6.3.18. Export online workflow states

The command export, in CSV format, the network data (load flow results) of all the states of an online workflow.

```
~/itesla_wp5/bin/itools export-online-workflow-states --workflow 20150625151301333 --file
~/itesla_wp5/states.csv
```

Options:
--workflow: id of the online workflow.
--file: csv file with all states.

Example of output (in standard and validation mode):

| workflow | state | cimName | datetime | daytime | month | forecastTime | horizon | SSCESL61SSCHR_ACLS__TO__SSCESP6_S_VL6_P | other op. conditions… |
|---|---|---|---|---|---|---|---|---|---|
| 20130115_1830_20151204131644694 | 0 | 20130115_1830_FO2_FR0 | Tue Jan 15 18:30:00 CET 2013 | 66600000 | 1 | 1470 | DACF | 134.02362 | … |
| 20130115_1830_20151204131644694 | 64 | 20130115_1830_FO2_FR0 | Tue Jan 15 18:30:00 CET 2013 | 66600000 | 1 | 1470 | DACF | 139.51735 | … |
| 20130115_1830_20151204131644694 | 1 | 20130115_1830_FO2_FR0 | Tue Jan 15 18:30:00 CET 2013 | 66600000 | 1 | 1470 | DACF | 122.018005 | … |
| 20130115_1830_20151204131644694 | 65 | 20130115_1830_FO2_FR0 | Tue Jan 15 18:30:00 CET 2013 | 66600000 | 1 | 1470 | DACF | 140.72894 | … |
| 20130115_1830_20151204131644694 | 2 | 20130115_1830_FO2_FR0 | Tue Jan 15 18:30:00 CET 2013 | 66600000 | 1 | 1470 | DACF | 151.8527 | … |
| 20130115_1830_20151204131644694 | 66 | 20130115_1830_FO2_FR0 | Tue Jan 15 18:30:00 CET 2013 | 66600000 | 1 | 1470 | DACF | 154.62936 | … |
| 20130115_1830_20151204131644694 | 3 | 20130115_1830_FO2_FR0 | Tue Jan 15 18:30:00 CET 2013 | 66600000 | 1 | 1470 | DACF | 119.76939 | … |
| 20130115_1830_20151204131644694 | 67 | 20130115_1830_FO2_FR0 | Tue Jan 15 18:30:00 CET 2013 | 66600000 | 1 | 1470 | DACF | 161.00987 | … |
| 20130115_1830_20151204131644694 | 4 | 20130115_1830_FO2_FR0 | Tue Jan 15 18:30:00 CET 2013 | 66600000 | 1 | 1470 | DACF | 139.60204 | … |

### 6.3.19. Run the security rules on a predefined online workflow state

The command runs the security rules on a stored state of an online workflow.

```
~/itesla_wp5/bin/itools run-security-rules-on-state --workflow 20150625151301333 --state 0 --offline-workflow 1358271900000-1356994800000-1359673140000
```

<u>Options:</u>
--workflow: id of the online workflow.
--state: id of the online workflow state.
--offline-workflow: id of the offline workflow that produces the security rules. If not provided, the offline workflow used for running the online workflow is used.
--wca: get results of WCA rules (uses MCLA rules if not specified).

Example of output (in standard and validation mode):

```
+-------------------+-------------+-----------+-------------+-----------------+------------------+----------------+--------------+
| Contingency       | SMALLSIGNAL | TRANSIENT | TSO_OVERLOAD | TSO_OVERVOLTAGE | TSO_UNDERVOLTAGE | TSO_SYNCHROLOSS | TSO_FREQUENCY |
+-------------------+-------------+-----------+-------------+-----------------+------------------+----------------+--------------+
| N-2_Tavel-Tama    | NOK         | NA        | NOK          | NOK             | NOK              | OK              | OK            |
| N-2_Waran         | OK          | NA        | OK           | NOK             | NOK              | OK              | OK            |
| N-1_Tavel-Tama    | OK          | NA        | OK           | NOK             | NOK              | OK              | OK            |
| N-2_Cergy-Terrier | OK          | NA        | OK           | NOK             | OK               | OK              | OK            |
| N-2_Launa_Taute   | NOK         | NA        | OK           | NOK             | NOK              | OK              | OK            |
| N-2_Avoi_Distr    | NOK         | NA        | OK           | OK              | NOK              | OK              | OK            |
| N-2_Rouge_Barna   | NOK         | NA        | OK           | NOK             | NOK              | OK              | OK            |
| N-2_Flama_Menu    | NOK         | NA        | OK           | NOK             | NOK              | OK              | OK            |
+-------------------+-------------+-----------+-------------+-----------------+------------------+----------------+--------------+
```

### 6.3.20. Run the impact analysis on a predefined online workflow state

The command runs an impact analysis on a stored state of an online workflow.

```
~/itesla_wp5/bin/itools run-impact-analysis-on-state --workflow 20150625151301333 --state 0
```

Options:

--workflow: id of the online workflow.

--state: id of the online workflow state.

--contingencies: contingencies to test, separated by “,” (all the db if not specified).

Example of output (in standard and validation mode):

```
+-------------------+----------+------------------+-------------+-----------+---------------+
| Contingency       | OVERLOAD | OVERUNDERVOLTAGE | SMALLSIGNAL | TRANSIENT | TSO_FREQUENCY |
+-------------------+----------+------------------+-------------+-----------+---------------+
| N-2_Cergy-Terrier | NOK      | NOK              | NOK         | OK        | OK            |
| N-2_Tavel-Tama    | NOK      | NOK              | NOK         | OK        | OK            |
| N-1_Tavel-Tama    | NOK      | NOK              | NOK         | OK        | OK            |
| N-2_Avoi_Distr    | NOK      | NOK              | OK          | OK        | OK            |
| N-2_Flama_Menu    | NOK      | NOK              | OK          | OK        | OK            |
| N-2_Launa_Taute   | NOK      | NOK              | OK          | OK        | OK            |
| N-2_Rouge_Barna   | NOK      | NOK              | OK          | OK        | OK            |
| N-2_Waran         | NOK      | NOK              | OK          | OK        | OK            |
+-------------------+----------+------------------+-------------+-----------+---------------+
```

### 6.3.21. Print metrics (debugging)

The command prints the metrics stored by the online workflow returned by some online steps.

```
~/itesla_wp5/bin/itools print-online-workflow-metrics --workflow 20150625151301333 --step STABILIZATION --state 0
```

Options:

--workflow: id of the online workflow.

--step: the online step (FORECAST_ERRORS_ANALYSIS, MERGING, WORST_CASE_APPROACH, MONTE_CARLO_SAMPLING, LOAD_FLOW, SECURITY_RULES_ASSESSMENT, CONTROL_ACTION_OPTIMIZATION, TIME_DOMAIN_SIMULATION, STABILIZATION, IMPACT_ANALYSIS).

--state: the state id (if not specified the metrics of all the states are printed, in CSV format).

The command can print the metrics related to a specific step (for now, just available for STABILIZATION and LOAD_FLOW) and a specific sample (in the example below, the stabilization module failed for state 4, and details of the error can be found in the metrics):

```
./itools print-online-workflow-metrics --workflow 20150917105739449 --step STABILIZATION --state 4
```

Example of output (in standard and validation mode):

```
+-------+-------------------+-------------------------------------------------------------------------------------------------+
| State |     Parameter     |                                             Value                                               |
+-------+-------------------+-------------------------------------------------------------------------------------------------+
|     4 | steadyStateErrors | [{"machine":"MANOS6GR","macrobloc":"RT_HYDRH","variable":"D(SREGUL  )/DT","blocdesort":"1","valeurPu |
|       |                   | ":19.4342},{"machine":"REVI57G2","macrobloc":"RT_HYDRH","variable":"D(SREGUL  )/DT","blocdesort":"1" |
|       |                   | ,"valeurPu":21.1003},{"machine":"REVI57GR","macrobloc":"RT_HYDRH","variable":"D(SREGUL  )/DT","blocd |
|       |                   | esort":"1","valeurPu":25.3751},{"machine":"REVI57G0","macrobloc":"RT_HYDRH","variable":"D(SREGUL  )/ |
|       |                   | DT","blocdesort":"1","valeurPu":27.2171},{"machine":"HERMI6G2","macrobloc":"RT_HYDRH","variable":"D( |
|       |                   | SREGUL  )/DT","blocdesort":"1","valeurPu":298.6122},{"machine":"COGE36A0","macrobloc":"AMFAREGU","va |
|       |                   | riable":"","blocdesort":"21","valeurPu":2396.6187}]                                                  |
|     4 | lf_diverge        | false                                                                                           |
|     4 | lf_iterations     | 7                                                                                               |
|     4 | stab_rich_status  | COMPLETED                                                                                       |
+-------+-------------------+-------------------------------------------------------------------------------------------------+
```

The same command can print (in csv format) the metric related to a step for all the states:

```
./itools print-online-workflow-metrics --workflow 20150917105739449 --step LOAD_FLOW
```

Example of output (in standard and validation mode):

```
state,cause,contraintes,csprMarcheForcee,dureeCalcul,nbIter,statut
0,0,29,0,0.07,13,OK
1,0,56,0,0.04,12,OK
2,0,36,0,0.04,10,OK
3,0,46,0,0.06,13,OK
4,0,70,0,0.06,14,OK
```

### 6.3.22. Print the stored violations of an online workflow

The online workflow has the option of storing, for a state (i.e. a Monte Carlo sampled state) and a specific online step, the network (voltage and current) violations. So far, the violations are stored after the pre-contingency loadflow step of the MCLA (i.e. the LOAD_FLOW step), for each Monte Carlo state (state 0 is the base case).

The stored violations of a workflow can be printed (as table or in csv format) using the command:

```
./itools print-online-workflow-violations --workflow 20150917105739449 --step LOAD_FLOW
```

<u>Options:</u>
--workflow: id of the online workflow.
--step: the online step (FORECAST_ERRORS_ANALYSIS, MERGING, WORST_CASE_APPROACH, MONTE_CARLO_SAMPLING, LOAD_FLOW, SECURITY_RULES_ASSESSMENT, CONTROL_ACTION_OPTIMIZATION, TIME_DOMAIN_SIMULATION).
--state: the state id.
--csv: print in CSV format, instead of tabular format.

The command can print the violations of all states/steps, of a specific state (using the state id) or of a specific step (using the step id). For now, this command is just available for the LOAD_FLOW step.

Example of output (in standard and validation mode):

```
+-------+-----------+--------------+-------------+----------+-------+
| State |   Step    |  Equipment   |    Type     |  Value   | Limit |
+-------+-----------+--------------+-------------+----------+-------+
|     0 | LOAD_FLOW | M_PONP7_S_VL7 | LOW_VOLTAGE |  387.2086 | 391.0 |
|     0 | LOAD_FLOW | PALUEP7_S_VL7 | HIGH_VOLTAGE | 420.01477 | 420.0 |
|     0 | LOAD_FLOW | PONTEP7_S_VL7 | LOW_VOLTAGE  | 387.20828 | 391.0 |
|     1 | LOAD_FLOW | ANTIBP6_S_VL6 | LOW_VOLTAGE  | 226.83783 | 227.0 |
|     1 | LOAD_FLOW | ARGE6P6_S_VL6 | LOW_VOLTAGE  | 225.90776 | 226.0 |
|     1 | LOAD_FLOW | ARGIAP6_S_VL6 | HIGH_VOLTAGE | 252.78702 | 245.0 |
|     1 | LOAD_FLOW | ARGIAP7_S_VL7 | HIGH_VOLTAGE | 421.42554 | 420.0 |
|     1 | LOAD_FLOW | BATIGP6_S_VL6 | LOW_VOLTAGE  | 225.41414 | 226.0 |
|     1 | LOAD_FLOW | BIANCP7_S_VL7 | LOW_VOLTAGE  | 384.09125 | 391.0 |
|     1 | LOAD_FLOW | BOCCAP6_S_VL6 | LOW_VOLTAGE  | 226.88432 | 227.0 |
|     1 | LOAD_FLOW | BRAUDP6_S_VL6 | HIGH_VOLTAGE | 245.19269 | 245.0 |
|     1 | LOAD_FLOW | B_CARP7_S_VL7 | LOW_VOLTAGE  | 383.63913 | 391.0 |
|     1 | LOAD_FLOW | CALA5P7_S_VL7 | LOW_VOLTAGE  | 394.40472 | 396.0 |
|     1 | LOAD_FLOW | CANTEP6_S_VL6 | HIGH_VOLTAGE | 246.23953 | 245.0 |
+-------+-----------+--------------+-------------+----------+-------+
```

### 6.3.23. Print post-contingency violations

The online workflow also stores, for each sample (and so, also for the base case), the post-contingency violations obtained from the computation of a post-contingency load flow.

The stored post-contingency violations can be printed using the command:

```
./itools print-online-workflow-postcontingency-violations --workflow 20150917105739449
```

Options:
--workflow: id of the online workflow.
--state: the state id.
--contingency: the contingency id.
--csv: print in CSV format, instead of tabular format.

The command can print the violations for a specific state (using the state id), for a specific contingency (using the contingency id), or for the whole workflow. The output can be printed in a table (see example below), or in csv format.

Example of output (in standard and validation mode):

| State | Contingency | Equipment | Type | Value | Limit |
|---|---|---|---|---|---|
| 0 | N-1_Tavel-Tama | M_PONP7_S_VL7 | LOW_VOLTAGE | 386.7887 | 391.0 |
| 0 | N-1_Tavel-Tama | PONTEP7_S_VL7 | LOW_VOLTAGE | 386.7884 | 391.0 |
| 0 | N-2_Avoi_Distr | AVOI5L61AVOIN_ACLS | CURRENT | 1250.7559 | 1115.0 |
| 0 | N-2_Avoi_Distr | AVOI5Y761_PT | CURRENT | 1250.7559 | 913.0 |
| 0 | N-2_Avoi_Distr | M_PONP7_S_VL7 | LOW_VOLTAGE | 387.2038 | 391.0 |
| 0 | N-2_Avoi_Distr | PONTEP7_S_VL7 | LOW_VOLTAGE | 387.20343 | 391.0 |
| 0 | N-2_Avoi_Distr | P_HA5P7_S_VL7 | LOW_VOLTAGE | 393.644 | 396.0 |
| 0 | N-2_Cergy-Terrier | CERGYP7_S_VL7 | LOW_VOLTAGE | 386.99976 | 396.0 |
| 0 | N-2_Cergy-Terrier | MEZE5P7_S_VL7 | LOW_VOLTAGE | 391.66116 | 396.0 |
| 0 | N-2_Cergy-Terrier | M_PONP7_S_VL7 | LOW_VOLTAGE | 387.19266 | 391.0 |
| 0 | N-2_Cergy-Terrier | PONTEP7_S_VL7 | LOW_VOLTAGE | 387.19235 | 391.0 |

### 6.3.24. Print post-contingency load flow violations

This command provides information about the convergence of the post-contingency load flow.

```
./itools print-online-workflow-postcontingency-loadflow --workflow 20150917105739449
```

Options:
--workflow: id of the online workflow.
--state: the state id.
--contingency: the contingency id.
--csv: print in CSV format, instead of tabular format.

The command can print the convergence status for a specific state (using the state id), for a specific contingency (using the contingency id), or for all state/contingency pairs. The output can be printed in a table (see example below), or in csv format.

Example of output (in standard and validation mode):

```
+-------+------------------+---------------------+
| State |    Contingency   | Loadflow Convergence |
+-------+------------------+---------------------+
|     0 | N-1_Tavel-Tama    | true                 |
|     0 | N-2_Avoi_Distr    | true                 |
|     0 | N-2_Cergy-Terrier | true                 |
|     0 | N-2_Flama_Menu    | true                 |
|     0 | N-2_Launa_Taute   | true                 |
|     0 | N-2_Rouge_Barna   | true                 |
|     0 | N-2_Tavel-Tama    | true                 |
|     0 | N-2_Waran         | true                 |
|     1 | N-1_Tavel-Tama    | true                 |
|     1 | N-2_Avoi_Distr    | true                 |
|     1 | N-2_Cergy-Terrier | true                 |
|     1 | N-2_Flama_Menu    | false                |
|     1 | N-2_Launa_Taute   | false                |
|     1 | N-2_Rouge_Barna   | true                 |
|     1 | N-2_Tavel-Tama    | true                 |
|     1 | N-2_Waran         | false                |
+-------+------------------+---------------------+
```

### 6.3.25. Run a dynamic simulation on a specific basecase (or a list of base cases)

This command will run dynamic simulations on a selected base case (from CIM repository) or a list of situations (all the ones contained in the `case-dir` folder and its subfolders), for a list of specified contingencies or without contingencies. If no contingencies are specified, the list is defined by the online configuration.

The workflow is not yet configurable to run without contingencies (i.e. just with the empty contingency). For now, to run the simulation just on the empty contingency, the online workflow configuration must be changed to point to an empty CSV contingencies file, and the `empty-contingency` parameter must be used.

```
./itools run-td-simulation-mpi --case-dir <DIR>
```

Options:
--case-dir: the directory where the case is.
--case-basename: the case base name (all cases of the directory if not set).
--contingencies: contingencies to test separated by , (all the db if not set).
--empty-contingency: include the empty contingency among the contingencies.
--output-folder: the folder where to store the data.

The output is stored in 3 files (saved in the `output-folder`, if specified, or in the folder where the command is running). The output files are the following:

- “networks-violations.csv”: it contains, for each base case, the pre-contingency violations.
- “simulation-results.csv”: it contains, for each base case, the results of the T-D simulation on all the contingencies (just the empty contingency, if the command is run with no contingencies).
- “metrics.log”: it contains, for each base case, the metrics, output of stabilization and impact analysis steps. It could be useful to understand why a stabilization step (i.e. a td-simulation with the empty contingency) failed.

#### 6.3.26. Run a FPF on a specific basecase

```
./itools run-fpf
```

Options:
--analysis: id of the forecast error analysis.
--time-horizon: time horizon (example DACF).
--base-case-date: base case date (example 2013-01-15T18:45:00+01:00).
--output-dir: output dir where the FPF output files will be stored.
--with-postprocessors: apply postprocessors to the network (step up transformers and javascript).

The analysis and time-horizon input parameters define which forecast errors models to use for the computation of the uncertainties used by FPF. If not specified, the command will use the parameters defined for the online workflow (online-default-parameters section in config.xml). In the output folder all the output files generated by FPF are stored. The folder will also contain a file with the mapping between IIDM ids (contingencies, buses, etc.) and the ids (numbers) referred by the FPF output files.

## 6.4. How is it possible to deal with an error message?

The logs of the online workflow can be found under /home/itesla/itesla_wp5/logs.

## 6.5. How is it possible to switch between the French network and the 7-bus network?

Two different networks coexist in the platform. The French network and the “overload” variant of the 7-bus test network which will be described later in the document. This is a temporary procedure for the validation phase which is why it is not “clean”.

The installation is currently configured to work with "France". To switch between the two configurations, you need to change the itesla.conf file in the etc folder (/home/itesla/itesla_wp5/etc).

Two already configured files can the found in the folder (itesla.conf_7buses_overload and itesla.conf_france). Use one of them (i.e. copy one of them to itesla.conf) in order to run the correspondent configuration. E.g. for working with 7buses:

```
cd /home/itesla/itesla_wp5/etc/
cp itesla.conf_7buses_overload itesla.conf
```

If any change is made in the configuration file, the platform has to be restarted with command:

```
cd /home/itesla/itesla_wp5/bin
./online-service.sh restart
```

### 6.6. How is it possible to switch between the HADES and HELM load flow method in the online workflow?

To use HELM, instead of HADES, as the load flow module in the online workflow, the configuration parameters must be changed in the config.xml file. In this file, in the `online` section and `loadFlowFactoryClass` tag, put `eu.itesla_project.helmflow.HelmFlowFactoryImpl` instead of `com.rte_france.itesla.hades2.Hades2Factory`, like:

```
<loadFlowFactoryClass>eu.itesla_project.helmflow.HelmFlowFactoryImpl</loadFlowFactoryC
lass>
```

The load flow is also used by the component that adds the Eurostag step up transformers to IIDM network. To configure the components to use HELM, you should, again in the config.xml file, in `EurostagStepUpTransformerPostProcessor` section and `loadFlowFactoryClass` tag, put `eu.itesla_project.helmflow.HelmFlowFactoryImpl` instead of `com.rte_france.itesla.hades2.Hades2Factory`, as presented above.

### 6.7. How is it possible to disable the step up transformers?

To disable the step up transformers, it is necessary to change the configuration parameters in the config.xml file. Thus, in the import section, instead of the follow configuration:

```
<import>
     <postProcessors>stepUpTransformers,javaScript</postProcessors>
</import>
```

the following instruction must appear:

```
<import>
     <postProcessors></postProcessors>
</import>
```

## 7. ANNEXES:

### 7.1. Configuration files for the offline part:

These configuration files are only valid for the French/Belgian network. The configuration files for the 7-bus network will be described later in the document.

#### 7.1.1. iTesla configuration files

All configuration files are in ~/.itesla (workspace root).
The folder contains a main config.xml file and Java properties files (set of configuration variables with a defined value).

**Important:** Any changes made in these files will only be considered after restarting the offline platform, by using the command:

```
offline-service.sh restart
```

### 7.1.2. Main configuration files

Almost all the configuration parameters can be changed in config.xml.

Here are examples of useful parameters:

**Tag <offline-default-start-parameters>**
Change “sampleperthread” to define the amount of samples in the set of samples treated at the same time when the command “start-offline-workflow” is launched.
Change “duration” to change the default calculation duration

**Tag <simulation-parameters>**
Change “branchFaultShortCircuitDuration” to adjust the fault duration for faults on lines (in seconds).
It can be interesting to change the duration of a short circuit for tests relative to loss of synchronism.

**Tag <csvcontingencydb>**
“csvFile” contains the path towards the CSV file containing the list of the contingencies.
If the validator wants to use a specific contingency list, he or she can create a new list and change the path accordingly.

**Tag <xml-validation-db>**
“directory” contains the path towards the IIDM XML files associated to the generated samples.
These are the IIDM XML files from the Validation DB. They can be used to generate Eurostag files.

**Tag <EurostagStepUpTransformerPostProcessor>**
“ddbPath” is the path to access .zip file containing the description of the step-up transformers.
This is the data related to step up transformers of the generators which are not represented in the CIM files but are necessary to run dynamic simulations.

**Tag <rulesbuilder>**
Change this part to configure the generation of the decision trees

To tune the indexes parameters, a dedicated file “wp43adapter.properties” is still used.

## 7.2. Configuration files for the online part:

These configuration files are valid for any network.

#### 7.2.1. iTesla configuration files

Two configurations have been prepared:

- "France": /home/itesla/itesla_wp5/france/ folder for .xml configuration file and data
- "7buses overload": /home/itesla/itesla_wp5/7buses_overload/ folder for .xml configuration file and data

#### 7.2.2. Main configuration files

Almost all the configuration parameters for the online part can be changed in the online config.xml file.

The config.xml file is in the .itesla folder (e.g. /home/itesla/itesla_wp5/france/.itesla for the "France" configuration).

Some interesting parameters are:

**Tag <online-default-parameters>**
This part contains the default parameters of the online workflow (all of these parameters can be given in input, and overwritten, via the command line).

**Tag <fea-parameters>**
This part contains the default parameters for the forecast errors analysis (all of these parameters can be given in input, and overwritten, via the command line).

**Tag <forecastErrorsAnalyzer>**
This part contains other default parameters for the forecast errors analysis (these parameters can only be modified in the config.xml file).

To be completed if necessary

<u>Additional information about the current configuration of the online part:</u>

The base cases of the French network used for the analysis can be found in /home/itesla/itesla_wp5/france/caserepo.

The network used for the 7buses overload can be found in /home/itesla/itesla_wp5/7buses_overload/cim.

The csv files containing the contingencies data are in the configuration folders.

The tmp folders (e.g. eurostag data) of the computations that use mpi (i.e. online workflow and fea) can be found in /home/itesla/itesla_wp5/tmp, while the tmp folders containing computations done using the local computation manager (e.g impact analysis using the specific itool command) are in /home/itesla/itesla_wp5/france/localtmp/ or /home/itesla/itesla_wp5/7buses_overload/localtmp/ (depending on the configuration used, see above).

### 7.3. Detailed content of the databases:

These databases are only valid for the French/Belgian network.

If some of these databases are shared with the 7-bus network, it will be precised.

Some databases are common to the offline and online part and some databases are specific.

### 7.3.1. CIM repository:

This is a file system organized database: SN + FO organized per format/date/country. There is only one file in the database for now. This is a temporary situation.

The databases are not common to the offline and online part for now but could be configured this way.

This database is not shared with the 7-bus network.

### 7.3.2. Historical DB:

The database is continuously fed by SN and FO files (from the CIM database). Time series oriented data (transposed compared to CIM data, i.e. time series for a given state variable). This database is not on the same ovh server than the other databases.

A simple way to retrieve historical data is to use what was developed by Imperial College for the validation of WP4.1 which makes historical data available as .mat files.

Otherwise, it is possible to retrieve data making queries (see annex 7.13).

The database is common to the offline and online part.

This database is shared with the 7-bus network.

### 7.3.3. Dynamic DB:

The database contains EUROSTAG dynamic data (.fri, .frm, .par, .pcp, .rcp for every generator for now) and references to lib MODELICA models. The same database is used on the 7-bus network.

The database is common to the offline and online part.

This database is shared with the 7-bus network.

### 7.3.4. Step-up transformers DB:

The database contains a description of step up transformers (static data) to enrich the initial static description contained in the CIM files.

The database is common to the offline and online part.

There are no step-up transformers on the 7-bus network so no need for such a database.

#### 7.3.5. Contingencies DB:

The databases contain a description of the contingencies. A database contains a list (.CSV format) of lines, transformers and generators on which faults are be simulated. The dynamic description of the fault (stabilization phase, fault duration etc.) is to be done in the configuration files.

The database are not common to the offline and online part for now but could be configured this way. They have to be consistent (e.g contain the same list of contingencies) for the online workflows to be compatible with the offline workflows.

This database is not shared with the 7-bus network.

#### 7.3.6. Simulation DB:

The database contains WP4.2 states and WP4.3 security indicators.
The information is written by the workflow during sampling/simulation step and read by the workflow during security rules computation.

The database is specific to the offline part.

This database is not shared with the 7-bus network.

#### 7.3.7. Validation DB:

The database contains full WP4.2 cases in IIDM XML format. The data can be loaded in Convergence[11] for further studies. Combined with the information contained in Dynamic DB, it allows the user to generate input files for Eurostag. Unlike Simulation DB, this database contains the network structure for each sample which can be used for validation purposes.

The database is specific to the offline part.

This database is not shared with the 7-bus network.

#### 7.3.8. Forecast error repository

The forecast error directory is defined in the online config.xml file as the “forecastErrorsDir” parameter.

For each error calculation ID, there are two files (forecast_errors_DACF.mat and a statistics_DACF.csv).

These files give raw data about the error calculation.

#### 7.3.9. Security rules DB:

The information contained in the database is written by the workflow during security rules computation and read by the online workflow.

[11] Software used by RTE/CORESO to run static simulations.

The database is common to the offline and online part.

This database is not shared with the 7-bus network.

#### 7.3.10. Online DB

The online DB is the component in charge of storing online data, i.e. all the data the online workflow and its modules need to store (and that are not already stored in other platform repositories), e.g. results of the online workflow modules (wca, mcla, t-d simulation, etc.), sampled states, metrics, etc.

It is the component that provides the data when outputs like the online workflows or the modules results are printed using the itools commands.

### 7.4. Upload historical data on the historical database:

First connect to the server hosting the historical database: ns311581.ovh.net.

You can do it with a SSH client (Putty) using the same password than with servers ns6375578.ovh.net, ns6375579.ovh.net or ns6375582.ovh.net or use command line "`ssh itesla@ns6375578.ovh.net`" from the test server you are using.

For example, to upload the CIM files from January 2015 in the "frbe/2015" database:

- Check that the folder /home/itesla/CIM/SN/2015/01/ contains the CIM files to upload

- Check that the right database is configured:
```
<histodb>
<histoDbSourceName>frbe/2015</histoDbSourceName>
```

- Use command line to upload January 2015 in DB frbe/2015:
```
curl -X POST -k -u itesla:itesl4
'https://localhost:8183/histodb/rest/itesladb/frbe/2015/itesla' --
data-urlencode 'dir=/home/itesla/CIM/SN/2015/01/'
```

- Define the baseline:
```
curl -X POST -k -u itesla:itesl4
'https://localhost:8183/histodb/rest/ frbe /2015/itesla/referenceCIM'
--data '/home/itesla/CIM/SN/2015/01/01/20150101_0000_SN4_BE0.zip'
```

  In the example above, the baseline is the CIM file '/home/itesla/CIM/SN/2015/01/01/20150101_0000_SN4_BE0.zip'. Any situation can be defined as a baseline because it is not used in the calculation but bugs occur if no situation is defined.

Please note that when new files are added to the historical database, they are added to the existing files (except if their name is the same than the existing ones in which case they are replaced).

## 7.5. Upload dynamic data to the dynamic database:

This part describes how to handle the dynamic database related to Eurostag.

Keep in mind that the dynamic database is common to servers 78, 79 and 82 for now. These commands have to be used only if some validators have a dedicated server.

The structure of the data is very specific and must be kept to upload new data:

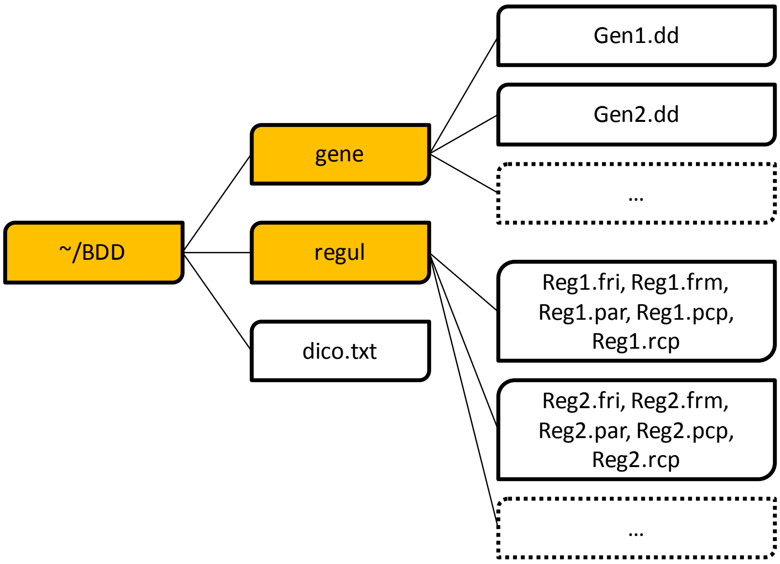


Figure 7 - 1: Structure of the DDB

- dico.txt contains the mapping between the CIM ID and the Eurostag ID.

  Example for the French network :

  CIM_ID;EUROSTAG_BDD_GEN
  ABBEV_TG1_WGU_SM;ABBEVT 1
  ABBEV_TG2_WGU_SM;ABBEVT 2

- gene is the folder in which the .dd files are stored. The .dd files are excerpts from the .dta file.

- reguls is a folder which contains the Eurostag regulations. The .fri .frm .par .pcp .rcp files must be stored for each regulation.

To update data to the dynamic database (e.g update data already existing in the database), the dynamic database must be **erased first and uploaded again!**

To erase the data base, use:

```
itools ddb-unload --host 0.0.0.0 --port 8080 --user user --password password
```

Parameters host, port, user and password must be consistent with the config.xml (tag <ddb>).

To upload the French data in the ddb, use:

```
itools ddb-load-eurostag --host 0.0.0.0 --port 8080 --user user --password password --eurostag-version 5.1.1 --data-dir ~/DDB
```

The ~/DDB folder must be structured as defined above.

It is possible to use `ddb-load-eurostag` a several times in a row to gradually upload data for different generators (but not update existing ones).

The Dymola dynamic database will be described later.

## 7.6. Data formats:

A specific data format internal to the platform "IIDM XML" (Internal iTesla Data Model XML) was developed for the needs of the project. These .xml files contain the whole description of the network. They can be opened with Convergence or Eurostag if combined with the dynamic data.

## 7.7. Fault pattern:

Two .seq files are generated to run the "stabilization" simulation and the "impact analysis" simulation. The pattern for the two simulations can be tuned in the config.xml file.

The fault patterns used in the platform are illustrated below:

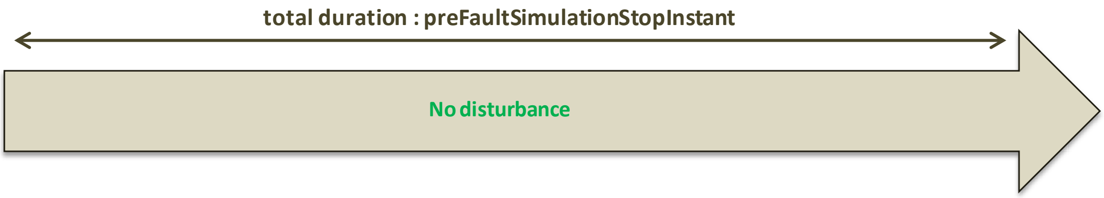


Figure 7 - 2: "stabilization" pattern

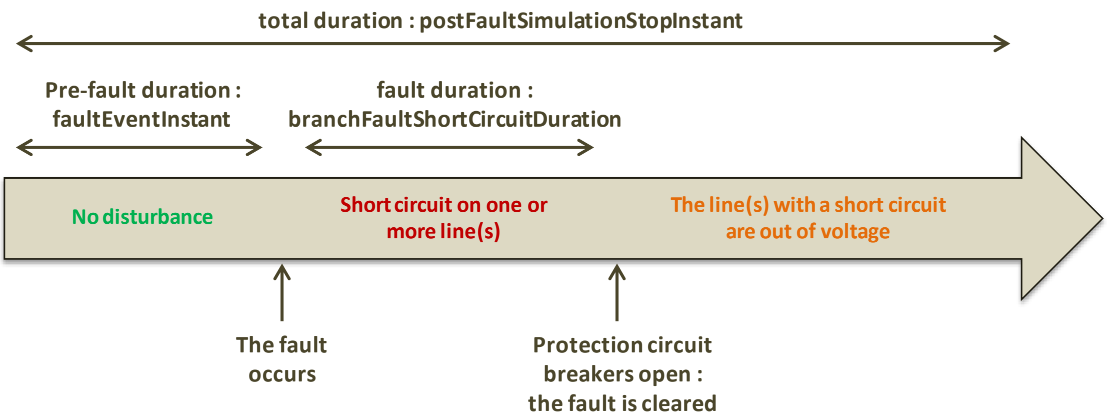


Figure 7 - 3: "impact analysis" pattern

For a line, the fault is simulated in the middle and is three-phased.

All the duration parameters in the config.xml file are in seconds.

To be adapted for generators

## 7.8. Units

The platform outputs use units adapted to the scale of transmission systems. The units are listed below:

| Physical quantity | Current | Voltage | Active Power | Reactive Power |
|---|---|---|---|---|
| Unit | A | kV | MW | MVAr |

## 7.9. Useful linux commands

The command-line interpreter or shell used is “bash”. Many online tutorials exist to learn the basic linux commands. Nevertheless, a few tips can help with the validation:

### 7.9.1. Useful linux particularities

Right click on a selection copies it where the cursor is.

CTRL+C combination kills a process.

### 7.9.2. How is it possible to copy/paste from windows to linux terminal and vice versa?

If you work in a windows environment and access iTesla platform through Putty for example, copying text from windows environment to the linux terminal and vice versa can be useful:

Windows to Linux

- Copy as usual in windows environment
- Select linux emulator
- Paste using right-click

Linux to Windows

- Copy using scroll wheel click in linux emulator
- Select windows environment
- Paste as usual in windows environment

### 7.9.3. How can intermediary results be saved for later?

If instead of just visualizing an output you want to save it in a text file, use:

```
~/itesla/bin/itools name-of-the-command [--name-of-option1 option1 --name-of-option2 option 2 ...]>/tmp/results.txt
```

#### 7.9.4. How is it possible to copy files from the linux server to another linux computer?

Use command:

```
scp itesla@ns6375578.ovh.net:/origin_folder/name_of_file_to_copy.csv /destination_folder
```

Change the server address if necessary.

For file transfers from linux environment to Windows environment, softwares such as WinSCP can also be very useful.

## 7.10. Configured indexes:

Only few indexes have been integrated on the platform so far. These indexes are:

- Smallsignal, Transient, Overload, OverUnderVoltage described in D4.3 (Definition of expected results from time domain simulations).
- TSO_Synchroloss, TSO_Overvoltage, TSO_Undervoltage, TSO_Overload, TSO_Frequency, TSO_generator_voltage_automaton, TSO_generator_speed_automaton developed by RTE.

Here is a short definition of the RTE indexes:

TSO_Overload compares the current in steady state (last value of current, it may be a problem is the steady state is not reached) on all the lines and transformers of the network with its operational upper limits as defined in the CIM file. If at least one limit is violated, the situation is considered unacceptable.
TSO_Overvoltage compares the voltage in steady state on all the buses of the network either with the upper limit defined in the CIM file or with the maximum value in the historical data. If at least one limit is violated, the situation is considered unacceptable.
TSO_Undervoltage compares the voltage in steady state on all the buses of the network either with the lower limit defined in the CIM file or with the minimum value in the historical data. If at least one limit is violated, the situation is considered unacceptable.
TSO_Frequency compares the frequencies of the 400kV buses with their barycentre. If at least 10 buses exceed the limit defined, the situation is considered unacceptable.
TSO_Synchroloss compares the rotoric angles of the generators with their barycentre. If the sum of the generators for which the rotoric angles deviate more than 360 degrees exceed 100 MW, the situation is considered unacceptable.
TSO_generator_voltage_automaton considers a situation is unacceptable if the voltage automaton created in Eurostag disconnects a generator. The voltage automaton disconnects a generator if the voltage at the connection point is under the lower threshold or above the upper threshold for a defined duration. Unlike the TSO_synchroloss index, no minimal power amount is defined.
TSO_generator_speed_automaton considers a situation is unacceptable if the speed automaton created in Eurostag disconnects a generator. The speed automaton disconnects a generator if its speed is under the lower

threshold or above the upper threshold for a defined duration. Unlike the TSO_synchroloss index, no minimal power amount is defined.

To tune the KTH indexes, use `wp43adapter.properties` file.

## 7.11. Specific user guide for the 7-bus network in the offline platform:

The 7-bus network was implemented on the same server than the French network. They co-exist on the platform for validation purposes only. Some choices in the architecture might seem inconsistent but were made for efficiency reasons. For example: there is no direct link between the file tree for the 7-bus and the DB structure. The file tree for the 7-bus was made in order to differentiate elements of the variants (contingencies, CIM base case etc.). In some cases, the DB are common to the 2 networks (Historical DB, Dynamic DB) in other cases (Contingencies DB, CIM repository DB) they are different either because they have to be changed by the validator or because the DB structure doesn't allow it.

The file tree with the variants, the specific command options, the useful configuration files and DB to be accessed and changed are specific to it and will be described below.

### 7.11.1. IDs

The buses are not simply named after their ID number. They have complex IDs which have to be taken into account for tests:

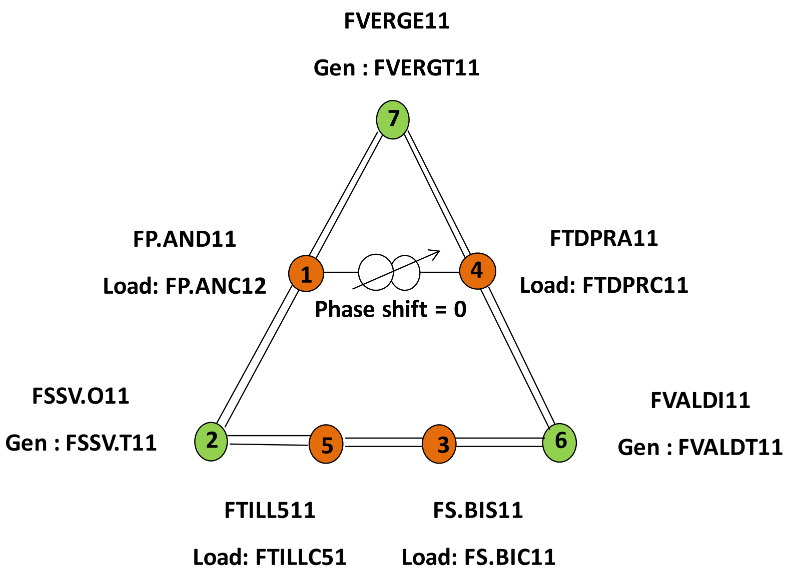


Figure 7 - 4: 7-bus IDs

In the representation above, the first ID associated with every bus corresponds to the bus ID itself. The "load" or "gen" IDs correspond to the load or the generator IDs connected to the buses.

### 7.11.2. File tree

The folder "7buses_test" gathers all the specific files needed for the tests on the 7-bus. The file tree is represented below. The sub files and sub folders are only represented once for "overload" folder but are the same for every variant.

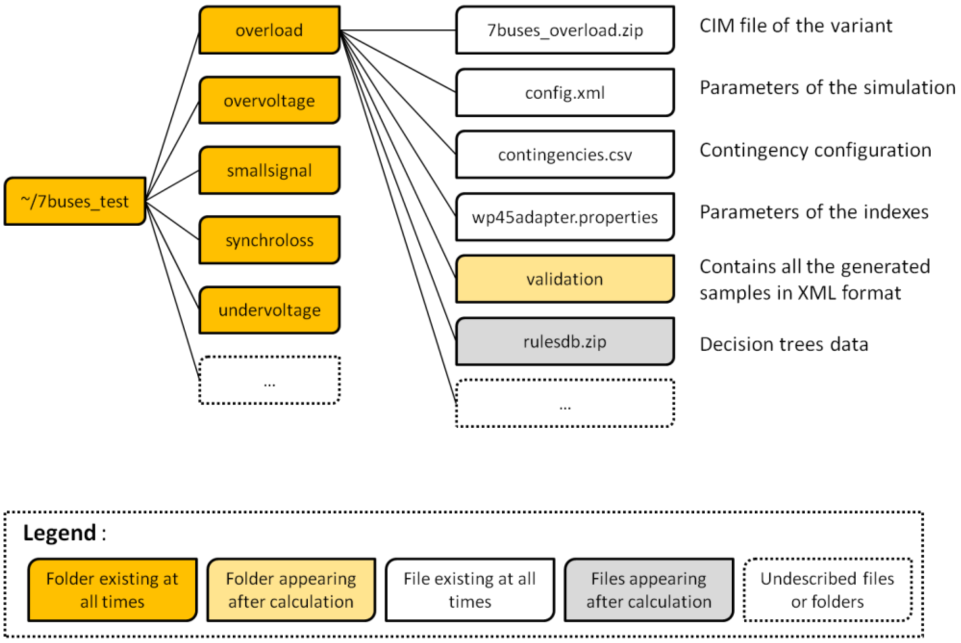


Figure 7 - 5: 7-bus file tree

### 7.11.3. Choice of the variant

Read paragraph 3.7 to make sure the configuration allows you to work on the 7-bus network.

To choose the variant, type command:

(1) `[itesla@ns6375578 ~]$ cd ~/itesla/etc`
(Go to the correct directory)

(2) `[itesla@ns6375578 etc]$ vi itesla.conf`
(Edit the configuration file. The bold part of the output below is to be changed with the name of the variant to study)

```
itesla_cache_dir=/home/itesla/7buses_test/undervoltage/cache
itesla_config_dir=/home/itesla/7buses_test/undervoltage
itesla_config_name=config
…
```

(3) `[itesla@ns6375578 ~]$ offline-service.sh restart`
(If anything was changed in the configuration, restart the service)

### 7.11.4. Specific databases to be changed

#### 7.11.4.1. Change the CIM basecase in the CIM DB

On the 7-bus network, the CIM database consists of one basecase CIM file in the folder associated with the variant. In the file tree example, the CIM database is the 7buses_overload.zip file.

For validation purpose, the original CIM file in the folder can be replaced by a new one provided that all the CIM files have the same name than the original zip file.

For example, if you want to replace the zip file 7buses_overload.zip, make sure that you create 3 files called 7buses_overload_EQ, 7buses_overload_SV and 7buses_overload_TP and compress them together in a zip file called 7buses_overload.zip.

#### 7.11.4.1. Change the contingency list in the contingencies DB

The contingencies DB is in the “contingencies.csv” file in the file tree example.

To change or display the contingency list associated with the overload index, use for example:

```
[itesla@ns6375578 overload]$ vi contingencies.csv
```

The pattern of this database for N facilities:

```
Name_of_contingency;N;Name_of_faulty_facility_1;…;Name_of_faulty_facility_N
```

<u>Define more parameters for a contingency:</u>

In some cases (loss of synchronism for example), it can be useful to define some of the contingencies precisely and differently from the default values.

The patterns of the faults can be specified in the file: ~/.itesla/simulationDetailedParameters.xml.

In the example below, a fault on “Tavel-Tamareau 400kV” with a duration of 0.151 is defined at 1% of the line from the substation “Tamareau 400kV”.

```
<?xml version="1.0" encoding="UTF-8"?>
<simulationDetailedParameters>
    <contingency id="N-1_Tavel-Tama">
           <branch    id="TAMARL71TAVEL_ACLS"    shortCircuitDuration="0.151"
shortCircuitDistance="1" shortCircuitSide="TAMARP7_S_VL7"/>
    </contingency>
</simulationDetailedParameters>
```

#### 7.11.4.2. Access the .xml files in the validation DB

The validation DB is in the “validation” folder in the file tree example. The .xml files associated with the workflow “workflow-0” are in the folder “validation/offline/workflow-0”. All the .xml files are zipped.

#### 7.11.4.3. Access the simulation DB

The simulation DB is in the folder “offlinedb” which is in the folder associated with the chosen variant. It is not represented in the file tree because the validator doesn’t have to access it directly.

#### 7.11.4.4. Access the .json files in the security rules DB

The security rules database is in the “rulesDB.zip” file in the file tree example. It contains all the decision trees in “json” format. They can be read in PEPITE software “DataMaestro”.

### 7.11.5. Commands for the 7-bus network

The commands listed in this document apply to the 7-bus network but options and start directory have to be adapted.

Here are examples of basic commands to launch tests on the 7-bus network and analyze results. This example was launched on the “UNDERVOLTAGE” variant.

(13) `[itesla@ns6375578 ~]$ itools list-offline-workflows`
(List all the existing workflows)

(14) `[itesla@ns6375578 ~]$ itools create-offline-workflow`
(Create offline workflow)

(15) `[itesla@ns6375578 ~]$ itools list-offline-workflows`
(Check that the workflow was created)

(16) `[itesla@ns6375578 ~]$ itools start-offline-workflow --workflow workflow-0`
(Start the sampling phase on workflow-0)

(17) `[itesla@ns6375578 ~]$ itools list-offline-workflows`
(Check that the workflow is running and wait for the 15mn to be over)

(18) `[itesla@ns6375578 ~]$ itools compute-security-rules --workflow workflow-0`
(Compute the security rules)

(19) `[itesla@ns6375578 ~]$ itools print-security-indexes-synthesis --workflow workflow-0`
(Print the security synthesis)

(20) `[itesla@ns6375578 ~]$ itools print-security-rule --workflow workflow-0 --attribute-set MONTE_CARLO --contingency FSSV_O1__FTILL51__1_ACLS --index-type TSO_UNDERVOLTAGE --purity-threshold 0.95 --format ASCII_FLAT`
(Prints a security rule)

(21) `[itesla@ns6375578 ~]$ itools export-security-indexes --workflow workflow-0 --output-file /tmp/offlinedb.csv --add-sample-column`
(Export validation .csv file in /tmp/offlinedb.csv)

(22) `[itesla@ns6375578 ~]$ itools export-metrics --workflow workflow-0 --output-file /tmp/metrics.csv`
(Export debug .csv file in /tmp/metrics.csv)

(23) `itools export-eurostag --case-dir ~/7buses_test/undervoltage/validation/offline/workflow-0 --case-basename sample-0 --case-format XML --output-dir /tmp/eurostag-0`
(Export eurotag files for sample 0 in /tmp/eurostag-0)

### 7.11.6. Configuration files

The 7-bus network is almost configured like the French network.

To tune the simulation parameters, use for example:
`[itesla@ns6375578 undervoltage]$ vi config.xml`
(Tune the simulation parameters for the undervoltage variant)

Ex:
Change “sampleperthread” to define the amount of samples in the set of samples treated at the same time when the command “start-offline-workflow” is launched.

To tune the indexes parameters, use for example:
```
[itesla@ns6375578 overvoltage]$ vi wp43adapter.properties
```
(Tune the indexes parameters for the overvoltage variant)
It’s the only part of the configuration which is defined as “XX.properties” and not in “config.xml”.

## 7.12. Imperial College WP4.1 validation module

### 7.12.1. Description of the module

A dedicated module has been developed in MATLAB to enable dataset comparison for validation purposes. In the context of WP4.1, this pertains to the comparison between historical and sampled stochastic data. In general, we expect the sampled dataset to exhibit the same statistical attributes as the dataset used to parameterize the truncated C-Vine model. Note that due to the high-dimensionality of the datasets, there is no single number that can accurately summarize dataset similarity. To this end, visual comparison of scatter plots aids in ensuring the two datasets have similar dependence structures. Since it is not possible to plot multivariate scatter plots, we resort to variable aggregation and variable selection. In the former, we plot and compare scatter plots of sums of variables. In the latter, two specific variables are chosen and plotted.

The visualCompator_v01 MATLAB function along with some example files can be downloaded from:
https://itesla.atlassian.net/wiki/display/ITESLA/WP4.1+-+Sampling+of+stochastic+parameters?preview=/2785364/41123842/visualComparator_v01.zip
Extract and add the visualCompator_v01.m file in your MATLAB path.

The visual comparator module is written in MATLAB and executed using the command visualComparator_v01(a,b,c,d), where:
a = full or relative location of first .mat file
b = full or relative location of second .mat file
c = first set of variables to be aggregated and compared
d = second set of variables to be aggregated and compared
Note that it is possible to forego supplying the c and d inputs i.e. visualComparator_v01(a,b). In this case, variables will be split in half for comparison. It is important to ensure that the two datasets have the same number of variables (columns) or an error message will be given.

For example, if we have the historical dataset 'Y_hist.mat' (1283 observation and 128 variables) and sampled dataset 'Y_sampled.mat' (2000 observations and 128 variables), then we can run a visual comparison using “half and half” variable aggregation by executing:
visualComparator_v01('Y_hist.mat','Y_sampled.mat',[1:64],[65:128]);
or equivalently
visualComparator_v01('Y_hist.mat','Y_sampled.mat');
Note that other splitting and variable combinations are valid such as
visualComparator_v01('Y_hist.mat','Y_sampled.mat',[1:4],[5:128]);

visualComparator_v01('Y_hist.mat','Y_sampled.mat',[1:4],[3:128]);
visualComparator_v01('Y_hist.mat','Y_sampled.mat',[1:4,67,89],[3:128]);
To visually compare two individual variables e.g. variable 1 and 2, execute:
visualComparator_v01('Y_hist.mat','Y_sampled.mat',1,2);
Example outputs of these comparisons are shown in next figure.

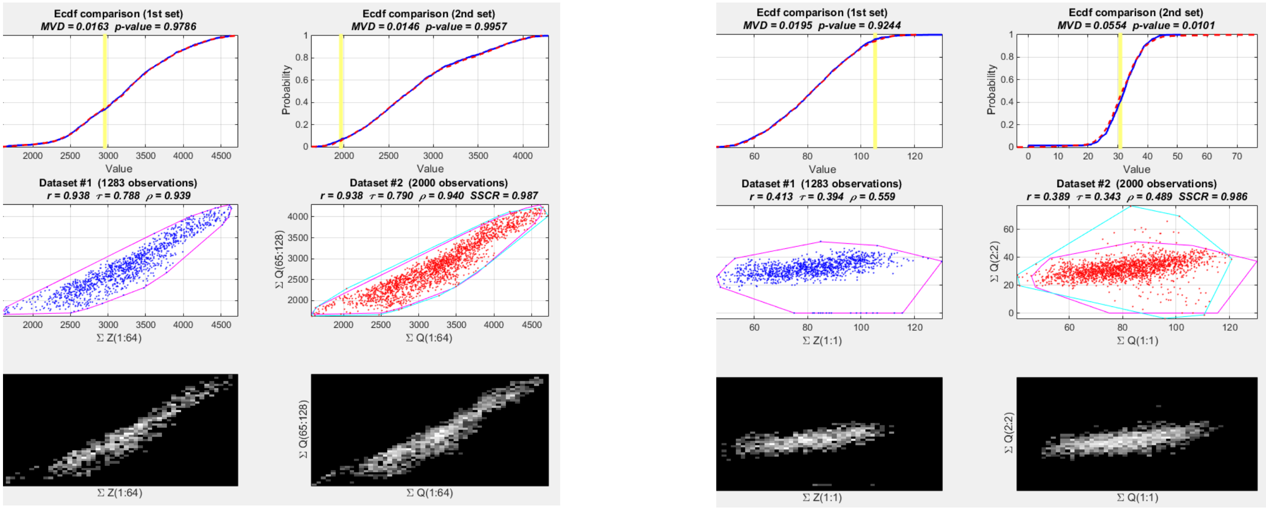


**Figure 7 - 6: Example outputs of the visual comparison module; variable aggregation (left) and single variable selection (right).**

As can be seen in the figures above, the output consists of a total of 6 six plots:

- The top two plots are ecdf (empirical cumulative distribution function) comparisons between the two datasets. More precisely, the leftmost top figure shows ecdf of dataset#1 (blue) and dataset#2 (dotted red) for the first set (e.g. summation of variables 1 to 64). The rightmost top figure shows ecdf of dataset#1 (blue) and dataset#2 (dotted red) for the second set (e.g. summation of variables 65 to 128). The blue and red curves should, ideally, match exactly. The vertical yellow line highlights the maximum deviation between the two (deviation magnitude is shown in the plot title as MVD – maximum vertical deviation). The Kolmogorov-Smirnoff test is used to extract a p-value from this comparison. This p-value can be interpreted as the likelihood that such an MVD can be observed if the two datasets have been obtained from the same underlying model. For a good match between historical and sampled data, this p-value must be high, ideally above 0.05. However, it is important to keep in mind that variable aggregation reduces the power of this test. When many variables are aggregated together, we would expect p-value to be much higher (e.g. > 0.1) in most cases. However, what is most important is to ensure that the ecdf shapes look similar. For example, looking at Figure 7 - 6, top right we can see that Kolmogorov-Smirnoff p-value is 0.01. However, given that the ecdf shapes look similar and occupy roughly the same range (highest density in the area 20-40), there is no significant reason to discard our statistical modelling.
- The two plots in the middle row are scatter plots. The leftmost figure (blue) shows scatter plot of dataset#1 while the rightmost figure (red) shows scatter plot for dataset#2. In this case, we want to ensure that the two figures look similar i.e. variables occupy the same area of the state space and dependence structure is largely the same. We also plot the convex hull of the datasets; pink for dataset#1 and cyan for dataset#2. This is to help detect outlying points that may not be easily visible. To assist this visual comparison, we also display three key attributes in the plot titles which should have roughly the same values. Note that when comparing between two individual variables, it is less likely that these statistical attributes will match very closely; when aggregating many variables together, the matching is expected to be much closer.
    - $r$ is the Pearson's correlation between the two datasets

- $\tau$ is the Kendall's tau (rank correlation) between the two datasets
- $\rho$ is Spearman's rho between the two datasets

- The two plots in the bottom are greyscale heat maps (similar to a 2-d histogram) to facilitate visual comparison in the case of a prohibitively large number of points in the afore-mentioned scatter plots. Color is related to point density, where black means low density and white means high density. Figure axis have been programmed to be the same in both plots to facilitate direct visual comparison. The user should ensure that the shapes between the two plots look similar (e.g. same angle, same areas of the state space occupied with roughly the same density of points).

### 7.12.2. How to use it in the platform

To be able to use the module, first edit the config.xml file associated with the current active variant (overload in the example below) and check that the following line is active (not between "<!--" and "-->"):

```
<validationDir>/tmp/WP41_overload</validationDir>
```

The folder between the two tags above is where the .mat outputs of the Imperial College module will be stored.

If any changes are to be made in the config.xml file, restart the platform using command:

```
offline-service.sh restart
```

Then, clean the cache using the script:

```
~/clean_cache.sh
```

There is no automatic way of emptying the output folder specified between the two "validationDir" tags. Before starting a new workflow, it is better to empty it manually.

Then create and start a new workflow.

When the calculation is finished, use command:

```
itools wp41-data-comparator --ofile /tmp/WP41_overload/compared_data --set1
[1,4] --set2 [5,8]
```

<u>Options:</u>
ofile: directory and name of the output .png and .fig files
set1: equivalent of "c" input defined in the previous paragraph
set2: equivalent of "d" input defined in the previous paragraph

This command is also particularly interesting because <u>it allows the validator to access the raw historical data</u> for active and reactive loads and the raw sampled data in the .mat files generated.

When the module is not needed, it is better to comment the line associated with tag <validationDir> in the configuration file in order not to generate useless files in the output folder.

## 7.13. Queries to the Historical DataBase

### 7.13.1. Introduction

Validators may have to retrieve data from the historical database to perform local tests. Thanks to a REST architecture (Representational State Transfer), this can be done directly through an HTTP request in a web browser. This documentation details the syntax to use to export data from the database and will be developed as long as new queries are made available.
For the moment, only queries on Historical Database are possible, to retrieve past and forecasted network states.

### 7.13.2. General syntax

An HTTP request to the database module should take the following form:

***https://server_address/path_to_rest_module/query_type?attributeA=valueA1,valueA2&...&attributeZ=valueZ***

All terms is *grey* have to be replaced by actual values:

- *server_address* is the DNS name of the iTesla server machine. Ex: ***ns311581.ovh.net***
- *path_to_rest_module* is the generic path to find the rest module on the server. Ex: ***histodb/rest/itesladb/france/2013full/itesla/***
- Other parameters are linked to a particular query and will be further explained.

An authentication to iTesla server may be requested to perform a query.
Here is a first example of a complete query:

***https://ns311581.ovh.net/histodb/rest/itesladb/france/2013full/itesla/data.csv?headers=true&equip=gen&attr=P,V&country=FR&count=-1&colRange=*&time=[2012-12-31T23:00Z,2013-03-30T22:59Z]&horizon=SN***

### 7.13.3. Query type

Different query types are used to choose if the query is to retrieve raw data of if a calculation has to be done on the result. The table below details the different functions currently available.

| Function | Query Type | Details |
|---|---|---|
| Raw data | ***data.csv*** | Retrieves raw data |
| Statistics | ***stats.zip*** | Retrieves statistics on the results : mean and 0.1%, 1%, 50%, 99%, 99.9% percentiles |
| Count | ***data/count.csv*** | Counts the number of results |
| Topologies | ***data/topos.csv*** | Automatic query to retrieve historical topologies for each substation. This query doesn't need any attribute. |
| Comparison SN-DACF or SN-D2CF | ***data/forecastsDiff.csv*** | Computes the difference between forecasted and realized network states in the results |

### 7.13.4. Attributes

Attributes are used to specify which network states have to be retrieved or will be used to perform the calculation, and which network properties have to be exported.

The syntax to be used is: ***attribute*=*value1*&*attribute*=*value1*,*value2***

Attributes are separated by **&**, values are affected to an attribute by **=** and different values for the same attribute are separated by **,**. The order of the different attributes is indifferent.

The following list details all the available attributes, the possible values for each, and some explanations on how to use them:

- ***async***
  - Possible values: ***true*** or ***false***
  - Can be used in all queries. Can be omitted.
  - Realizes an asynchronous calculation in order not to overload the server when much data is requested. Usually used in a statistic query. Run the same query to see the status.
- ***attr***
  - Possible values:

| Possible values | Categories | Units of the result | Possible values | Categories | Units of the result |
|---|---|---|---|---|---|
| ***P*** | Active power | MW | ***QP*** | Positive reactive power | MVar |
| ***V*** | Voltage | kV | ***QN*** | Negative reactive power | MVar |
| ***Q*** | Reactive power | MVar | ***T*** or ***TOPO*** | Substations topology | Lists |
| ***I*** | Current | A | ***TOPOHASH*** | Substations topology hash | Text |
| ***IP*** | Current / active power | A/MW | ***PGEN*** | Sum of active power production per station | MW |
| ***RTC*** | Tap of tap changers | - | ***QGEN*** | Sum of reactive power production per substation | MVar |
| ***PTC*** | Tap of phase shift transformers (both sides) | - | ***PLOAD*** | Sum of active power load per substation | MW |
| ***PP*** | Positive active power | MW | ***QLOAD*** | Sum of reactive power load per substation | MVar |
| ***PN*** | Negative active power | MW | ***QSHUNT*** | Sum of shunts and capacitors per substation | MVar |

  - Can be used in all queries.
  - Limits the variables used to specified categories
- ***colrange***
  - Possible values: * or any positive integer
  - Can be used in all queries.
  - Limits the number of columns in the results.
- ***cols***
  - Possible values: any ***column_name***. * can be used to replace any part of the name in order to filter several columns.
  - Can be used in all queries.
  - Specifies on which column the count has to be done. If a column name is specified, network states with the value NULL in this column won't be counted.
- ***count***

    - Possible values: ***-1*** or any positive integer
    - Can be used in all queries.
    - Limits the number of lines in the results.
- ***country***
    - Possible values: ***FR*** or ***...***
    - Can be used in all queries. Can be omitted.
    - Limits the location of the network properties processed (exported or used in a calculation)
- ***equip***
    - Possible values: ***gen*** or ***loads*** or ***stations*** or ***2wt*** or ***3wt*** or ***lines*** or ***dangling*** or ***...***
    - Can be used in all queries.
    - Limits the variables used to specified equipments: generators (***gen***), loads (***loads***), substations (***stations***), transformers with 2 or 3 windings (***2wt*** and ***3wt***), lines (***lines***), cut lines at borders of the network (***dangling***).
- ***filters***
    - Possible values: ***column_name:[min,max]***
    - Can be used in all queries. ***column_name*** doesn't have to be present in the attribute ***cols***. Can be used several times in the same query to filter several columns but different filters can't be separated only with a comma like other values.
    - Restricts network states on which the count has to be done with an interval of values for a particular variable. The variable ***column_name*** has to be between ***min*** and ***max*** for the network state to be used. ***min*** or ***max*** can be a decimal number (***0.01***) or ***Infinity*** or ***-Infinity***
- ***headers***
    - Possible values: ***true*** or ***false***
    - Can be used in all queries. Can be omitted.
    - If set to true, inserts the name of all exported columns and adds columns "datetime", "horizon" and "forecastTime" to the results
- ***horizon***
    - Possible values: ***SN*** or ***DACF*** or ***D2CF*** or ***FO***
    - Can be used in all queries. For a comparison query, the forecasted horizon is defined.
    - Restricts the network state used to a specified horizon of forecast. SN stands for SnapShot (realized state), DACF for Day Ahead Congestion Forecast and D2CF for 2 Days Ahead Congestion Forecast.
- ***time***
    - Format: ***[YYYY-MM-DDThh:mmzzzzz,YYYY-MM-DDThh:mmzzzzz]***
    - Example : ***[2012-12-31T23:00Z,2013-03-30T22:59Z]***
    - Can be used in all queries.
    - Restricts the network states used to a time interval defined by two dates in an ISO 8601[12] compatible format. For a forecasted network state, the horizon date is considered.

[12] See http://en.wikipedia.org/wiki/ISO_8601 for more details

### 7.13.5. Examples

| Queries | Details |
| --- | --- |
| **https://ns311581.ovh.net/histodb/rest/itesladb/france/2013full/itesla/ data.csv? headers=true &equip=gen &attr=P,V &country=FR &count=-1 &colRange=* &time=[2012-12-31T23:00Z,2013-03-30T22:59Z] &horizon=SN** | Retrieves raw data, active power and voltage of all French generators for states in the time interval [23h UTC 12/31/2012 – 22h59 UTC 03/30/2013] Only snapshots are considered. |
| **https://ns311581.ovh.net/histodb/rest/itesladb/france/2013full/itesla/ stats.zip? headers=true &equip=stations &attr=V &country=FR &count=-1 &colRange=* &time=[2012-12-31T23:00Z,2013-03-30T22:59Z] &horizon=SN &async=true** | Retrieves statistics (mean and percentiles) of voltage in French substations in the time interval [23h UTC 12/31/2012 – 22h59 UTC 03/30/2013] Only snapshots are considered. The calculation is done in asynchronous mode. |
| **https://ns311581.ovh.net/histodb/rest/itesladb/france/2013full/itesla/ data/count.csv? headers=true &cols=AMARG_TG3_WGU_SM_P &filters=AMARG_TG3_WGU_SM_P:[0.01,Infinity] &count=-1 &time=[2012-12-31T23:00Z,2013-03-31T22:59Z] &horizon=SN** | Counts in how many network states the generator ***AMARG_TG3_WGU_SM*** has an active power in the interval [0.01, infinity), in the time interval [23h UTC 12/31/2012 – 22h59 UTC 03/30/2013]. Only snapshots are considered. |
| **https://ns311581.ovh.net/histodb/rest/itesladb/france/2013full/itesla/data/topos.csv** | Retrieves all historical substations topologies |

| Queries | Details |
|---|---|
| **https://ns311581.ovh.net/histodb/rest/itesladb/france/2013full/itesla/ data/forecastsDiff.csv? headers=true &equip=loads &attr=P &country=FR &count=-1 &colRange=* &time=[2012-12-31T23:00Z,2013-03-31T22:59Z] &horizon=DACF** | Realizes the difference of all active power French loads between snapshots and DACF network states in the time interval [23h UTC 12/31/2012 – 22h59 UTC 03/30/2013]. |

**END**