iPST project

2017-2018 Roadmap

Contact: info@itesla-pst.org
Website: http://www.itesla-pst.org/

Plan

→ **Partners and organization**

→ **Current state of the project**

→ **Past year activities**

→ **2017-2018 roadmap**

Partners

1 TSO: Rte Réseau de transport d'électricité

5 industrials: Artelys optimization solutions; GRUPO AIA; TRACTEBEL Engineering GDF SUEZ; techrain; Pepite sa

4 research centers: RSE Ricerca Sistema Energetico; INESCTEC TECHNOLOGY & SCIENCE ASSOCIATE LABORATORY COORDINATED BY INESCPORTO PORTUGAL; Imperial College London; KTH VETENSKAP OCH KONST

The consortium involves several IT providers besides RTE in order for iPST users not to be dependent on a single company.

Organization

- **The iPST project code is open-source with a commercial-friendly license (MPL V2)**
  - Available on https://github.com/itesla/
  - Obligation:
    - To publish the modifications done to the code in order to be able to distribute it.
    - To provide the source code in the distributions.
- **The partners decide on the roadmap in order to guarantee the consistence of the iPST project.**
  - There is no need to be a partner to contribute (through github).
  - Once a year, a governance meeting allows new partner to enter.

Current state: FP7 repository split into 3 parts (nothing lost)

➔ **ipst-core**

- iIDM format, iIDM-XML imp/exporters.
- Distributed computation (Open MPI, HPC).
- Load-flow API
- Security analysis API

➔ **ipst-entsoe**

- UCTE import and CIM 14 import
- ENTSOE case repository
- Network merge

➔ **ipst:** the remaining FP7 code, among which is "living":

- iIDM to tabular format exporter
- Eurostag exporter (static and dynamic data)
- HistoDB server and client
- Online workflow (Advanced security analysis)

Current state: other repositories

→ **Ansible repository:**

- to obtain the scripts that will download the other repositories and install a developer platform

→ **ipsl: the iTesla Power System Library:**

- no new development, but maintained.

→ **RTE-MSDS:**

- Eurostag synchronous machine (P,Q,V) steady state domain computation

Past year activities

Developments during the past year

➔ ipst-core

- Non-functional improvement:
  - Better build process (Travis / Coveralls)
  - Unit tests to improve coverage
  - Compilation and install scripts
  - iTools commands uniformization
- iIDM format and API improvement
  - Extension management
  - Static Var Compensator
  - Line temporary limit handling (TATL: 5 mn, 20 mn…)
  - PST active control mode
  - HVDC lines

Developments during the past year

➔ ipst-core

- Security analysis interface (in order to interface with a load-flow that can compute the result of many contingencies at once)
  - ■ With filtering of pre-contingency violations
  - ■ With automatic default list*
  - ■ With iTesla Action Language for remedial action*
- Groovy scripts to design workflows (to post-treat results)
- Load-flow validation functionalities

➔ ipst-entsoe

- Unit tests of the UCTE importer
- CIM V14 anonymizer.

*currently in a specific branch

Developments during the past year

→ ipst

- Worst-case approach (robust security analysis) improvement.
- Eurostag export:
  - Static Var Compensator
  - Alignment with iIDM (Hades) Load-Flow results for a better warm-start initialization
  - Better node naming strategy
- Online security analysis process:
  - Better handling of pre-contingency violations
- FP7 Forecast Error Analysis code (by RSE)
- FP7 HistoDB dataserver (REST API depending on some Pepite private jars)

Developments during the past year

- ➔ **PSM: Power Systems on Modelica: a modelica based simulator**
  - To be open-sourced soon
  - With a GUI
  - Working on small systems (less than 100 nodes).
  - With OpenModelica and Dymola
  - With an XML model database
  - Compatible with iPSL
- ➔ **Ansible to install the development platform**
- ➔ **Nordic44 data (CIM V14 files) was released by KTH under GPL license:**
  - https://github.com/SmarTS-Lab/Nordic44-Nordpool
  - Download scripts are included in Ansible

Not achieved targets

→ Developments

- CIM CGMES importer and exporter:
  - Developments have not been started yet, but should start soon.
- iIDM-json format
  - Abandoned
- C++ iIDM-XML importer/exporter
  - Developed by RTE, but lack of use case for open-sourcing.
- Offline workflow matlab-based decision tree not released
  - No expressed interest in the open-sourcing.

→ Usability

- No tutorials made.
- No user trainings organized.
- FP7 handbook not reorganized.
- No binaries available.

Communication events

- ➔ **PSCC Genoa and CIGRE Paris**
  - Cf. http://www.itesla-project.eu/
- ➔ **PMAPS Beijing (INESC, RTE, RSE, IC, TechRain)**

  "Online Security Assessment with Load and Renewable Generation Uncertainty : the iTesla Project Approach"
- ➔ **Codit 2017 (INESC, RTE, RSE)**

  "Managing forecast uncertainty in power system security assessment”
- ➔ **EPCC (Pepite)**
- ➔ **Modelica conference on PSM (RTE)**
- ➔ **Internal Imperial College symposium**
- ➔ **IEEE GM 2017 (RTE)**

  "Convex Modeling of the Steady State Domain of Power Generators”

2017-2018 Roadmap

2017-2018 targeted use cases

→ For an academic or a computation module provider

→ For a TSO

→ For editors / ENTSOE (Load-flow validation)

Academic use case

- ➔ An academic is collaborating with a TSO.
- ➔ The TSO provides files in CIM format in a node-breaker oriented way.
- ➔ The academic uses the platform to convert it to a simpler bus-branch view and exports it to the iIDM tabular format.
- ➔ The academic can develop experimental code around the grid data.

TSO use case

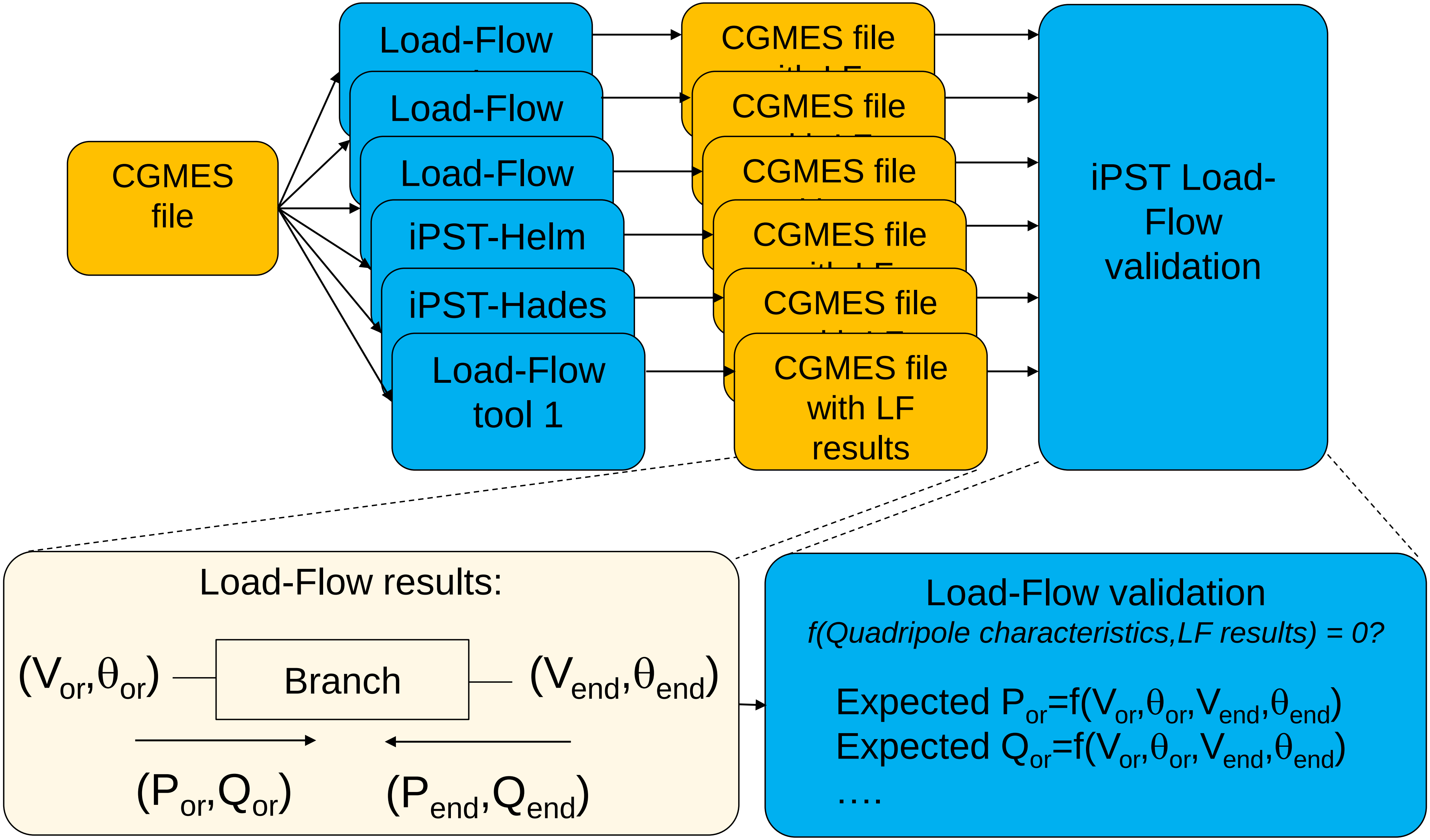
→ **Industrial implementation of standard security analysis:**

- Input: One grid state + A list of contingencies (implicit or explicit)
- Output: list of constraint violations
  - Undervoltage, overvoltage, overloads (including temporary admissible ones)
  - In pre- or post-contingency violations
- Underlying load-flow: Hades (freeware) or Helm

→ **Improvement expected:**

- JSON output format
- REST API.
- Curative remedial action simulations with iTesla Action Language.

Load-Flow validation use case



CGMES file

Load-Flow

Load-Flow

Load-Flow

iPST-Helm

iPST-Hades

Load-Flow tool 1

CGMES file

CGMES file

CGMES file

CGMES file

CGMES file

CGMES file with LF results

iPST Load-Flow validation

Load-Flow results:

(V_{or},θ_{or}) — Branch — (V_{end},θ_{end})

(P_{or},Q_{or}) (P_{end},Q_{end})

Load-Flow validation

f(Quadripole characteristics,LF results) = 0?

Expected $P_{or}=f(V_{or},\theta_{or},V_{end},\theta_{end})$

Expected $Q_{or}=f(V_{or},\theta_{or},V_{end},\theta_{end})$

….

iPST

Load-flow validation use case

- ➔ The Load-Flow validation tool is not a Load-Flow computation tool.
  - It only checks that the LF results are valid with respect to the LF problem definition.
- ➔ Validation allows to identify specific grid elements that are modelled differently by the LF tool and the validator.
  - The iPST Load-Flow validator is open-source and therefore allows to solve discrepancies, whereas it is difficult with closed source codes (and only paper documentation).
- ➔ Current code includes only checks on branches
  - Others checks on P/V generators, DC lines and slack node (in basic mode without distributed slack node) will be added.
- ➔ Targets:
  - ENTSOE interoperability tests (July 2018),
  - Large editor willing to check its “LF + CGMES export” tool chain.
  - Smaller editor willing to check its LF tool directly plugged into the platform through the “LoadFlow” API.

2017-2018 developments

- → **CGMES importer / exporter**
  - ● Goal: participate to the ENTSOE interoperability tests in July 2018
- → **iTesla Action Language for remedial action simulation.**
- → **Forecast error analysis model improvements**
  - ● Embedded into the Monte-Carlo like security analysis

Take-home messages

➔ **iPST: an open-source toolkit dedicated to large power grid simulations and security analysis**

➔ **Use cases exist for academics, TSOs and editors.**

➔ **Partners welcome!**