Powsybl basics, IIDM

iPST/PowSyBl day, 2018-05-25

Sylvain Leclerc
sylvain.leclerc@rte-france.com

Powsybl-core: basics

What it's not:

- A ready-to-run application (but … https://github.com/powsybl/powsybl-gse)

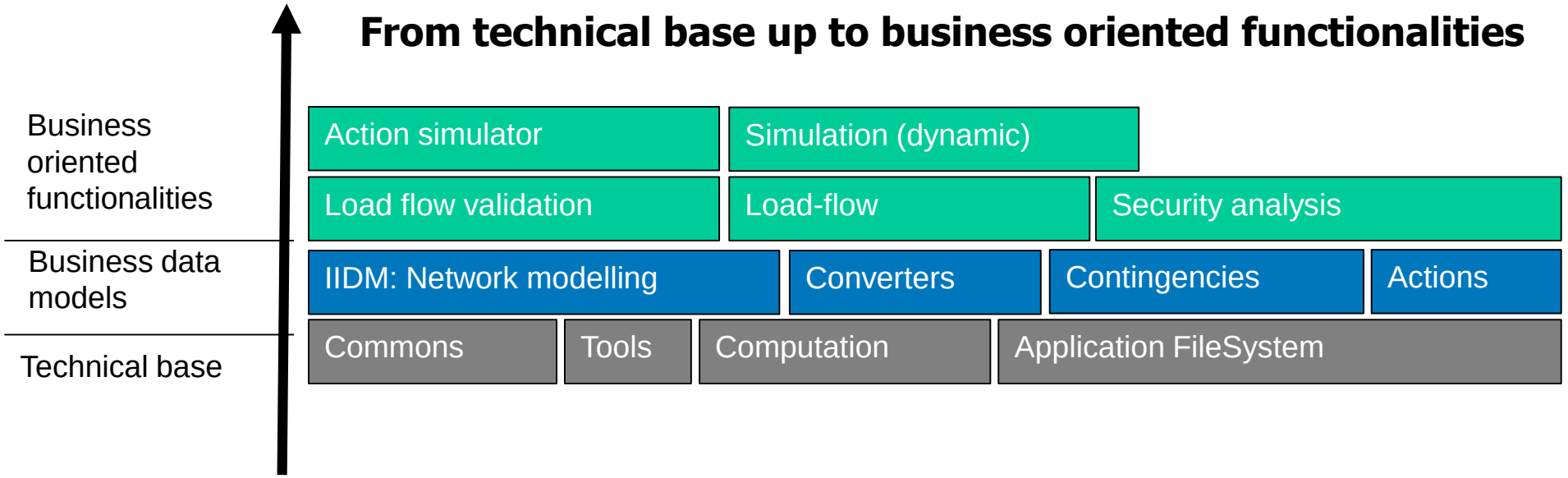
What it is :

- A set of mature libraries (mainly java 8) available to build your own power system analysis applications
- Comes with a command line tool to experiment/prototype/carry out specific studies

How to use it ?

- Download and compile (https://github.com/powsybl/powsybl-core)
- Include release as a maven dependency (current release: 1.0.0, ttp://www.mvnrepository.com/artifact/com.powsybl)

From technical base up to business oriented functionalities



Powsybl basics: IIDM

→ Itesla Internal Data Model

- It is the Network model used in the application
- Comes with:
  - An interface (~100 java classes, mainly pure interfaces)
  - One implementation of that interface
  - → *Opens for "non breaking" refactorings, or even your alternative implementation*
  - Small tests cases
  - Import/export from files

- Models for:
  - Substation
  - "Voltage level" in a substation
  - Switches
  - Substation topologies:
    - Bus / bars
    - Node / breaker
  - Lines
  - Two-winding transformers
  - Three-winding transformers
  - Phase-shifters and tap-changers
  - Loads
  - Generators
  - SVCs (Static VAR compensators)
  - Shunt compensators
  - HVDC lines
  - AC/DC conversion stations (LLC and VSC)

- Interface pretty well described in the Javadoc : http://www.itesla-pst.org/javadoc/powsybl-core/index.html?overview-tree.html

IIDM : usage

➔ **Among goals of the API:**

- Make it easy to build the network model
- Make it easy to use the network model
- Make it hard to "break" the network model (have an inconsistent model)

- Examples:
  - Building Eurostag example 1 network

IIDM : converters

- **Network models may be read from/written to files (or more generally streams of bytes)**
- **Converter API: "Importer" and "Exporter" interface**
- **Powsybl-core comes with implementations:**
  - XML file format : "XIIDM", image of the network model
  - UCTE format importer
  - ***Coming soon: CGMES importer/exporter***
- **Importer to be used is deduced from extension → easily use any kind of file**
- **Write and use your own converters through the plugin mechanism**
  1. Write an implementation of Importer interface
  2. Declare it as a service implementation with @AutoService annotation
  3. Put your compiled jar in execution classpath
  4. Use your own files !

Zoom on plugins

➔ **How can I add handling of my own format ??**

➔ **Lot of "client" behaviour is discovered at runtime**

- Use of @AutoService annotation from google
- Example 1: tools, adding new "itools" subcommands
- Example 2: Importers, adding new "itools" subcommands

➔ **You pick what you want to deploy for your application and what you don't**

IIDM: versioning

- **Network interface may evolve between powsybl-core releases, but we try to minimize it:**
  - Exemple: float to double for computation, in next or subsequent release

- **XML format may evolve between powsybl-core releases:**
  - Versioning of the format: currently 1.0
  - But what about my stored data ??
    Backward compatibility is guaranteed:
    - File 1.0 → powsybl 1.1 import then export → file 1.1
  - Note: may be actual file format changes, or interpretation changes

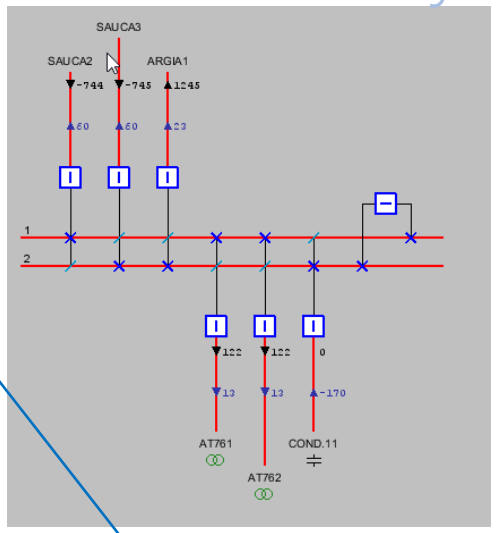
IIDM : scriptability

➔ **The model is fully accessible for scripting in groovy language**

➔ **Applications:**

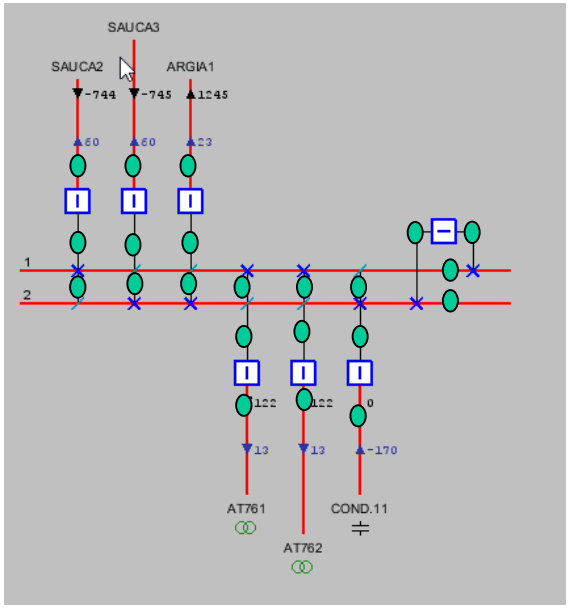
- Run simple computations without having to actually compile code.
  Ex: compute losses, …
- Post-process network model import before further business processing.
  Ex: run loadflow, change some setpoints, …
- Embed scripting possibilities in applications for advanced users
- Define custom “actions” in the “action DSL”
- …

IIDM : topological levels

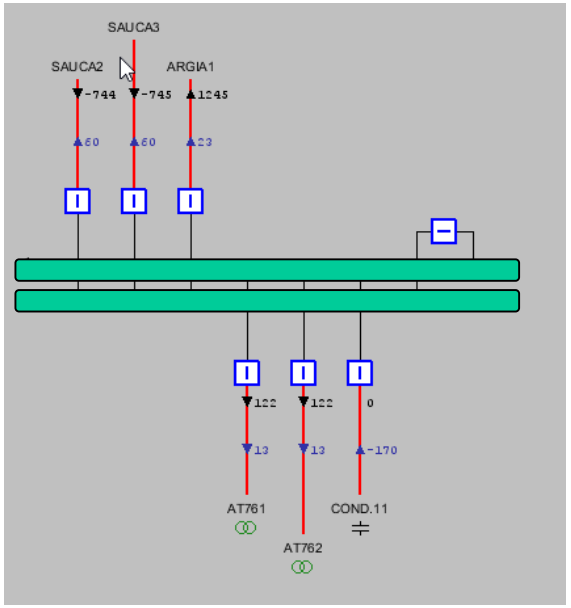


➔ IIDM can handle several levels of topology description:

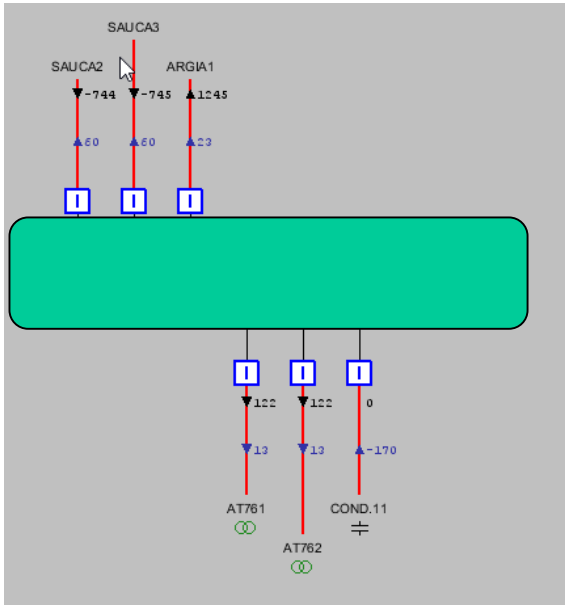
- Different views are accessible in the model, in particular for "voltage levels"
- The model itself may be built as "node-breaker" or "bus-breaker"
- Can convert node-breaker file to bus-breaker file



Nodes in « node-breaker » view, all switches

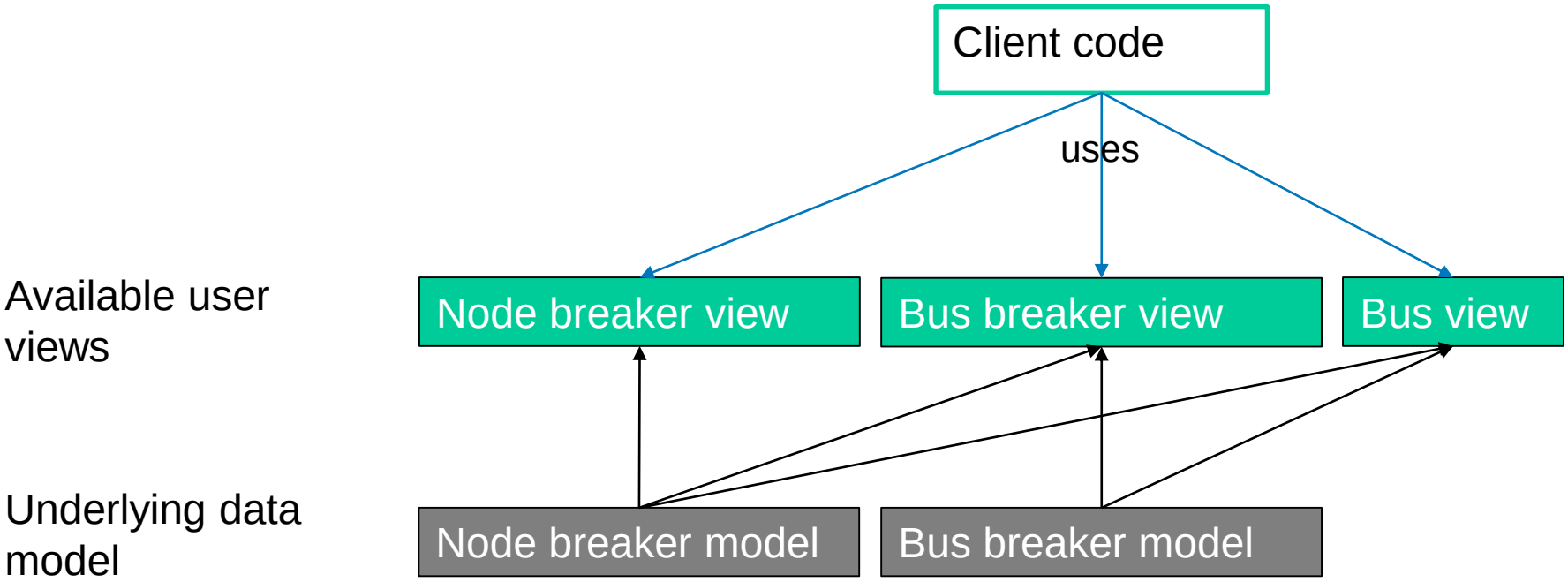


Buses in « bus-breaker » view, fewer switches



Buses in « bus » view, no more switches

Topological levels summary



➔ **Important to "natively" speak the 2 languages:**

- UCTE format → bus breaker model
- CGMES bus breaker or bus branch → bus breaker model
- CGMES node breaker → node breaker model
- …

IIDM : extensibility

➔ **All specifics of all networks cannot be planned in the "common" model**

➔ **The network model is easily extendable through the widely-used mechanism of plugins:**

- Every network equipment can carry extension objects
- Needs:
  - One class that represents your data
  - One class to read/write your data from/to xiidm

➔ **Exemples:**

- At RTE: "Stand-by automaton" of SVCs, HVDC specific regulation, generators frequency regulation …
- Merged X node extension: retain data associated to the 2 half lines of a merged network (CGMs)

Conclusions

➔ Powsybl-core:

- Set of production-grade java libraries (github/maven) to build power system applications
- Command line tool
- Frequent releases
- You can contribute
- Use it for your own needs ☺ (Commercial-friendly license MPL 2.0)

➔ IIDM:

- Computation-oriented network model
- Documented, easy to use, robust API
- Versioned format, backward compatibility
- CGMES support coming
- Easily pluggable in-house importers and exporters
- Extensible
- Handles different levels of details