TradeRES

New Markets Design & Models for 100% Renewable Power Systems

D4.8 – Open-access tool for linking electricity market models

Deliverable number:	*D4.8*
Work Package:	*WP4 – Development of Open-access Market Simulation Models and Tools*
Lead Beneficiary:	*VTT Technical Research Centre of Finland Ltd*

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 864276

Author(s) information (alphabetical)		
Name	**Organisation**	**Email**
Hugo Algarvio	LNEG	hugo.algarvio@lneg.pt
Niina Helistö	VTT	niina.helisto@vtt.fi
Juha Kiviluoma	VTT	juha.kiviluoma@vtt.fi
Ingrid Sanchez Jimenez	TUD	I.J.SanchezJimenez@tudelft.nl
Gabriel Santos	ISEP	gjs@isep.ipp.pt
Christoph Schimeczek	DLR	christoph.schimeczek@dlr.de
Ni Wang	TNO	ni.wang@tno.nl

Acknowledgements/Contributions		
Name	**Organisation**	**Email**

Document information			
Edition	**Date**	**Dissemination Level**	**Description**
1.0	25/11/2021	Public	Final
2.0	22/1/2024	Public	Final

Review and approval		
Prepared by	**Reviewed by**	**Approved by**
Niina Helistö (VTT)	Laurens de Vries (TUD)	Ana Estanqueiro (LNEG)

Disclaimer

The views expressed in this document are the sole responsibility of the authors and do not necessarily reflect the views or position of the European Commission or the Innovation and Network Executive Agency. Neither the authors nor the TradeRES consortium are responsible for the use which might be made of the information contained in here.

Table of Contents

List of Tables

List of Figures

1. Introduction

1.1 Goals for Task 4.3, Subtask 4.3.3 and Deliverable 4.8

Task 4.3: Integration of models with a model-linkage toolbox

Linking multiple models that specialise in certain aspects of the electricity markets allows to create more holistic and more robust simulation results. Works of model linkage in this task will be strongly connected to model updates from Tasks 4.1 and 4.2 and feedback from Tasks 6.1 and 6.2. Therefore, the work in this task will be ongoing and proceed even after delivering a first version of the multi-modelling toolbox in milestone M4.1.

Subtask 4.3.3: Integration with a model linkage toolbox

Herein we create a holistic model-linkage architecture for multi-modelling of energy markets exploiting existing platforms like Mosaik, VirGIL, Spine Toolbox and RCE. Once a platform is selected, we will integrate the partaking models in the platform. This will serve as an open-access market trading tool for expert stakeholders.

Deliverable 4.3: Open-access tool for linking electricity market models

A software framework of open-access electricity market models will be developed in this deliverable. Delivery month: M21 – first version (postponed to M22), M41 – final updated version (postponed to M48). (Originally D4.3)

1.2 Background

The Spine Toolbox[1] was chosen as the platform for integrating modelling tools. It is a free software application for managing data, scenarios and workflows for modelling and simulation tasks. The application is a result of the H2020 project Spine (2017–2021) and is in active development in the new EU project Mopo (2023–2036). The source code of Spine Toolbox is available under the GNU Lesser General Public License version 3 (LGPLv3).

With the Spine Toolbox, the user can create a workflow containing data items, data processing and validation, data management using a local or shared database, execution of modelling tools and archiving results including simple visualisations. These workflows are stored as projects and can be shared between users – or users can build their own workflows from the workflow components. The workflows are typically acyclic, but they can contain loops that allow to iterate between models, for example. In other words, a project contains the data processing and modelling tasks needed to accomplish certain research or other work.

The Spine Toolbox was chosen because of its strong workflow support and database-centred architecture as explained in Deliverable 4.7. (Cvetkovic et al., 2020).

[1] https://github.com/Spine-project/Spine-Toolbox

1.3 Scope and structure of the document

The deliverable is the set of Spine Toolbox projects that demonstrate the integration of various modelling tools to form an open-access tool for linking electricity market models. The scope of this document is a description of the Spine Toolbox projects (Section 2) and the shared datasets (Section 3).

2. Integration of models

The integration of the modelling tools that are used in TradeRES into the Spine Toolbox is happening one by one. First, each model is tested on its own by creating a Spine Toolbox project where the particular model is executed. Next, relevant models will be combined together. Table 1 shows the status and online repository location for each modelling tool. They are discussed in more detail in the following sub-sections.

Table 1: Status of modelling tool integration

Modelling tools	Status	Repository (under https://github.com/TradeRES)	Case study
Backbone	Completed	TradeRES-Backbone-demo	Europe
AMIRIS	Completed	toolbox-amiris-demo	Germany
EMLabpy + COMPETES-TNO	Completed	Spine_EMLab_COMPETES	-
EMLabpy + AMIRIS	In progress	toolbox-amiris-emlab	The Netherlands
MASCEM	In progress	mascem-demo	Iberia
Restrade + MASCEM	In progress	mascem-restrade-demo	Iberia

2.1 Backbone

A demo project for executing the Backbone modelling tool using the Spine Toolbox has been created. The dataset used is a simple representation of the German power system. The model is used to first find optimal investments and then find the operation strategy that yields minimum operational costs.

The Toolbox workflow for the project is shown in Figure 1. Original data files are connected through the item **BB_input_Excel**, **EV_data** and **Buildings** and the data is imported into the **BB_Spine_DB_direct** datastore.

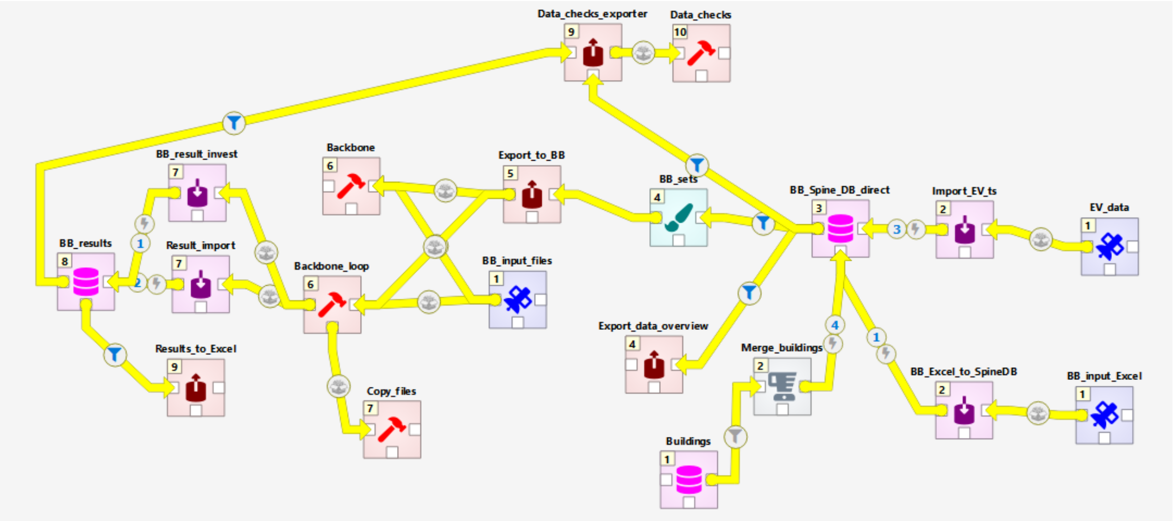

Figure 1: Spine Toolbox Backbone workflow

The workflow items **BB_sets** and **Export_to_BB** are used to create a binary GDX (GAMS Data eXchange) file which can be consumed by the model code. Backbone modelling tool, including investment and operational optimization stages, is executed in item **Backbone_loop**, and some additional configuration files are given by data connection **BB_input_files**. After running the model, results are collected into **BB_results** datastore using items **BB_result_invest** and **Result_import**. The results can be viewed in Excel format by running **Results_to_Excel**. The workflow includes also other items which can be used to copy the raw result files into summary folders, test the model, view an overview of the data and check the data for inconsistencies.

2.2 AMIRIS

The **A**gent-based **M**arket model for the **I**nvestigation of **R**enewable and **I**ntegrated energy **S**ystems (see, e.g. Deissenroth et al., 2017) is one of the four agent-based models within the TradeRES project. It is based on FAME[2], the open Framework for distributed Agent-based Modelling of Energy system. FAME simulations need to provide their input in a Protobuf[3] format and provide their output in a Protobuf file as well. Thus, to run an AMIRIS simulation, the Python-base tool FAME-Io is required to translate human-readable data to FAME-compatible files and vice versa.

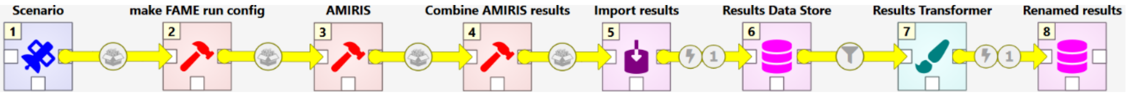

Figure 2: Workflow for AMIRIS execution within Spine Toolbox

[2] https://gitlab.com/fame-framework

[3] https://developers.google.com/protocol-buffers

Figure 2 shows the workflow for the execution of an AMIRIS simulation run including the following steps: First, the input data in YAML[4] and CSV format are translated into a single file in Protobuf format utilising FAME-Io[5] (step ‘make FAME run config’). The protobuf file is directed to “AMIRIS”, which is given as JAR[6]-container and must be executed on any Java Virtual Machine supporting Java 11 or higher versions. After a few seconds, the simulation is finished, and its output file can be converted to files in CSV format in step (‘Combine AMIRIS results’). To simplify further content assessment, the output files’ content is read and joined to a data frame utilising the wide-spread library Pandas[7] in that same step. The resulting data frame can then be imported into the Spine database ‘Results Data Store’, from which the symbols can be extracted, renamed and further transformed to any required format.

Please check out the FAME-Core and FAME-Io ReadMe files and FAME Wiki[8] for detailed instructions and troubleshooting advice.

2.3 EMLabpy and COMPETES-TNO

The first prototype of model coupling with the Spine Toolbox involved the models EMLabpy and COMPETES-TNO. This coupling is made in support of the Dutch national case study of WP5. The coupled model is used to explore the investment trajectory in new generation assets in the presence of additional revenues from the capacity market. The CO_2 market model is included in the analysis. The investment considerations are made yearly, and hence, this coupling operates with a time resolution of one year. The time horizon can be specified by user. EMLabpy modules provide capacity mechanism and CO_2 market functionalities, while COMPETES-TNO provides accurate information about operational costs in the current year and is additionally used to support investment decisions by providing valuation of different technologies in a future year.

[4] https://yaml.org/

[5] https://gitlab.com/fame-framework/fame-io

[6] https://docs.oracle.com/javase/tutorial/deployment/jar/

[7] https://pandas.pydata.org/

[8] https://gitlab.com/fame-framework/wiki/-/wikis/home

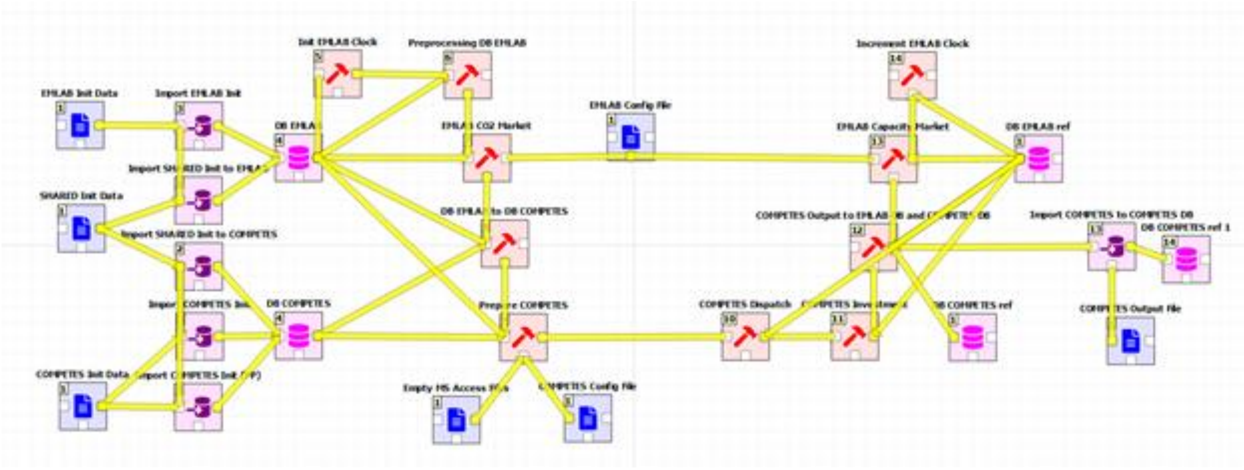

Figure 3: Workflow for coupling of EMLabpy and COMPETES -TNO within Spine Toolbox

The Toolbox workflow is shown in Figure 3. This workflow represents one iteration of a larger simulation, in which all iterations repeat the same steps. The Spine Toolbox provides looping capabilities which allow us to consider the workflow in Figure 3 as one iteration. One iteration represents one year, i.e., the investment decision for the current year is taken within the workflow in Figure 3.

The workflow in Figure 3 will be explained from left to right. In this implementation, three input files with initialization data are envisioned, one of which is a common dataset shared by both models. The datasets are imported to databases of EMLabpy and COMPETES - TNO. The so-called **EMLabpy clock** is used to initialize and increment the iteration mechanism, leading to the looping functionality. The **EMLabpy CO2 Market** module calculates the costs and revenues from the CO_2 market for all generation companies. The **COMPETES Dispatch** item runs the electricity system model of the current year, calculating operational costs and revenues. The **COMPETES Investment** item decides which new assets to install, and these are made available (and become implemented in the **COMPETES Dispatch** item) considering their respective building times. Next, the **EMLabpy Capacity Market** calculates the capacity market clearing price. The capacity market remuneration is considered for the new investments and decisions on decommissioning. The final step in the workflow is to save the results into databases of COMPETES-TNO and EMLabpy so that these become the initial state for the next iteration.

For more details about this workflow, please refer to Hommes (2021).

2.4 EMLabpy and AMIRIS

EMLabpy and AMIRIS are Agent Based Models (ABMs) with complementary capabilities. EMLabpy has the capability to simulate investments based on a future market expectation. EMLabpy is based on EMLab, which has the capability to simulate a short-term market but with limited flexibility. On the other side, AMIRIS models micro-economically strategic dispatch decisions and can simulate flexible resources, providing market results to EMLabpy, which interprets those as actors' expectations. Conducting a co-simulation of both models allows investigating market designs with an ABM approach. Figure 5 depicts the co-simulation architecture for both models, where the upper figure is the workflow in Spine Toolbox, while its abstract representation is shown in the bottom diagram.

In contrast to soft-linking models, in a co-simulation the inputs and outputs are exchanged for a specific time step, and based on the information the simulation progresses to the next step. Linking models involves utilizing the output of one model as the input for another model without requiring time synchronization.

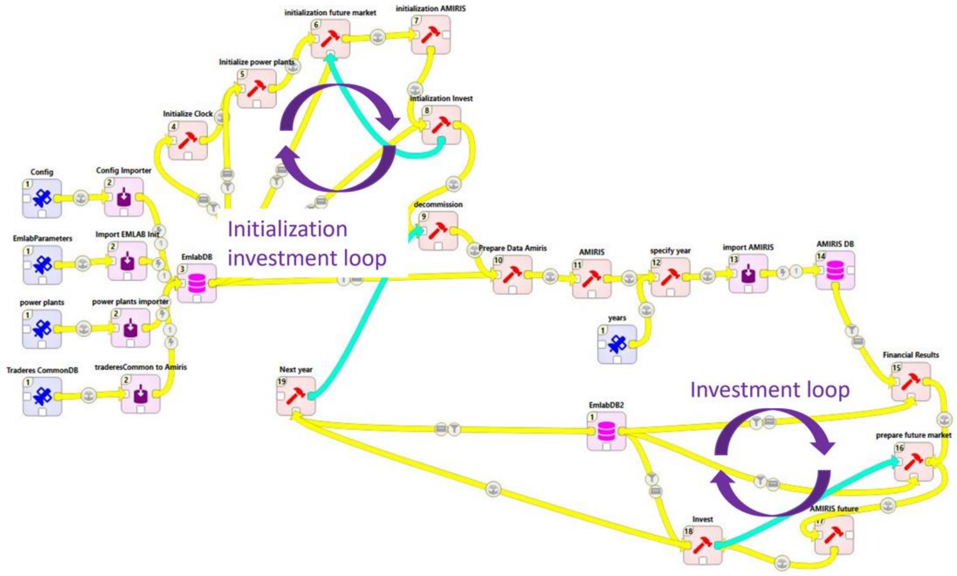

Figure 4: a)): Workflow EMLabpy-AMIRIS. Spine Toolbox workflow.

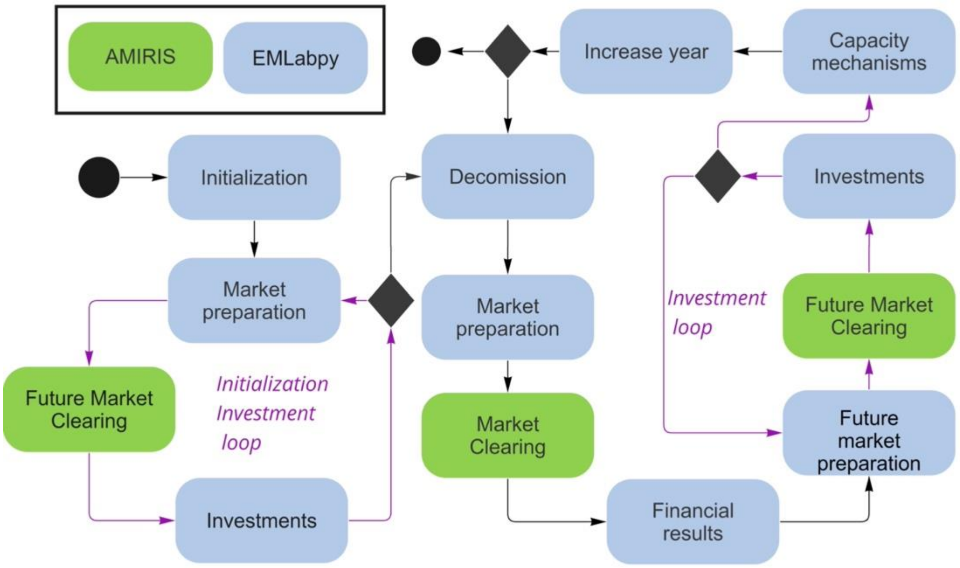

Figure 5: b)): Workflow EMLabpy-AMIRIS. Conceptualized workflow.

In a first step, an initial investment loop finds the missing installed electricity generation capacity from an initial set of power plants. After this, the yearly loop starts. Every simulation year begins by incrementing the age of the power plants. Subsequently. the operational power plants are filtered and passed to AMIRIS that clears the market. Next, the financial results are computed on a per power plant basis. Then, in the yearly investment loop, the market is cleared four years in advance. The investment decisions are based on the outcomes of AMIRIS. In the final step, the capacity mechanism market is calculated. Finally, the cycle proceeds to the next year or to the simulation end.

This co-simulation has been used to simulate the different scenarios for the Dutch case study.

2.5 MASCEM

The Multi-Agent Simulator for Competitive Electricity Markets (MASCEM) (Santos et al., 2016) is a modelling and simulation tool designed to study complex restructured electricity market operations by modelling the dynamic market players, including their interactions and the collection of medium/long-term data and experience, to support participants in making decisions based on to their characteristics and goals.

This project provides the implementation of MASCEM's electricity market models and power flow models in Spine Toolbox. It is prepared to represent the Iberian electricity market – MIBEL[9] and to validate the power network constraints, according to the market results. The Spine Toolbox workflow is shown in Figure 6.

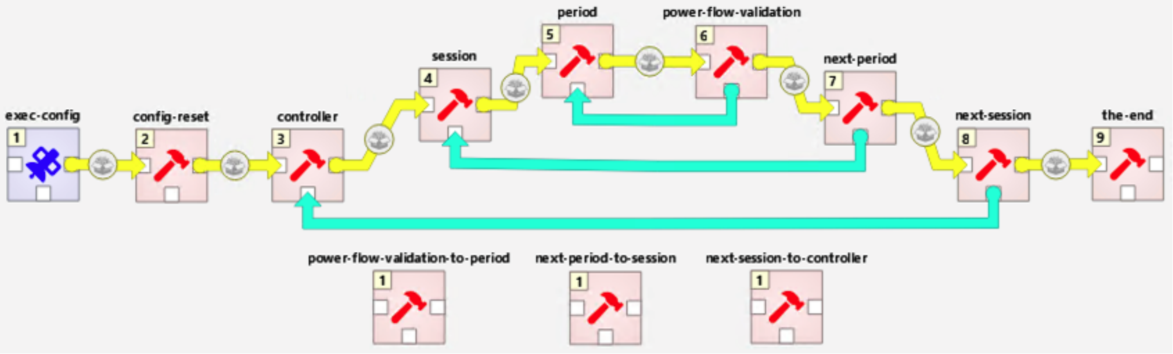

Figure 6: Workflow of MASCEM's Spine Toolbox project

The **exec-config** item (left-hand side) connects the configuration file for the simulation to the remaining items.

Table 2 presents the respective properties and possible values.

[9] https://www.mibel.com/en/home_en/

Table 2. Simulation configuration.

Property	Possible values
market_design	0 (day-ahead)
	1 (6hr-ahead)
simulation_scenario	0 (day-ahead-base)
	1 (day-ahead-forecasts)
	2 (6hr-ahead-forecasts)
ems.force_exec	true | false (default)
pfs.run	true (default) | false
pfs.dlr.PT->ES (MVA)	5748 (spring)
	3334 (summer)
	5748 (autumn)
	9836 (winter)
pfs.dlr.ES->PT (MVA)	4790 (spring)
	2778 (summer)
	4790 (autumn)
	8197 (winter)

The project's root folder includes a directory ("in”) where input files must be placed. During the execution, the user may stop the simulation at any time. The **config-reset** component ensures that auxiliary files generated to control the simulation are reset so it can run smoothly from the top.

The **controller** component is responsible for the control of the simulation, controlling the proper sessions' execution. In turn, the **session** component ensures the proper execution of the respective session's periods. The **period** item runs the Mibel pool for the respective period. Afterwards, the **power-flow-validation** item verifies if the period's clearing is feasible, according to the period's season dynamic line rating (DLR) and energy flow direction (i.e., PT->ES, ES->PT). If not, market splitting occurs, players are separated by bidding area, and the period pool is rerun for each area independently (PT and ES). This is ensured by the **power-flow-validation-to-period** tool through the blue connection between the **power-flow-validation** and **period** components.

If the power flow is valid, the **next-period** tool increments the period to be run next. If it is not the last period, the **session** tool is rerun (ensured by the **next-period-to-session** tool). Otherwise, the session is concluded, a session output file is generated with results tables and charts, and the **next-session** tool is executed, performing similarly to the **next-period** component (in this case, ensured by the **next-session-to-controller** tool).

Finally, the **the-end** tool concludes the simulation removing all temporary auxiliary data project's directory.

2.6 RESTrade and MASCEM

This project integrates selected model capabilities from MASCEM and RESTrade, namely by integrating MASCEM's wholesale day-ahead market models with RESTrade's secondary and tertiary energy markets.

In the current stage of development, the RESTrade system has models of secondary capacity procurement and imbalance settlement. It also includes models for secondary capacity and energy markets, such as for the tertiary energy market. This integration aims to enable the study of the effect of high penetrations of vRES (variable Renewable Energy Sources) in short-term (balancing) markets and identify potential changes to their market designs. Against this background, the market models of MASCEM and RESTrade have been improved to deal with closer to real-time closures and smaller time steps (15 minutes). Furthermore, the intraday continuous market has also been developed and integrated into MASCEM and will be available soon in this coupled version. Concerning RESTrade, the dynamic procurement of reserves has been integrated such as market coupling and a new Imbalance Settlement mechanism to incentivize self-balancing.

In the current stage of development, the Toolbox workflow of the coupled MASCEM and RESTrade models is as presented in Figure 7.

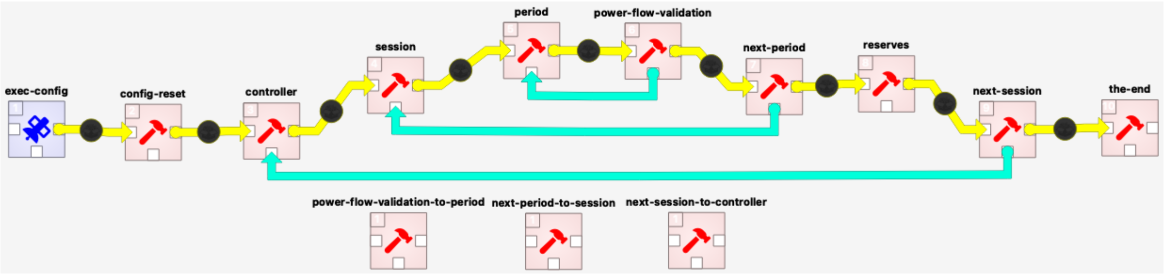

Figure 7: Workflow of Spine Toolbox project comprising the integration of MASCEM with RESTrade

The **reserves** item between next-period and next-session items contains RESTrade-specific tool configurations. The remaining items of the workflow are presented in the previous subsection about MASCEM.

The **reserves** item contains models of

- the procurement of secondary capacity,
- the secondary capacity and energy markets,
- the tertiary energy market, and
- the Imbalance Settlement.

Users may select either marginal pricing or the pay-as-bid scheme to clear these markets in the config.txt file, as indicated in the user guide of the RESTrade system (RESTrade User Guide). Furthermore, in the same file, users may select if they want to simulate traditional or new market designs. New market designs consider: i) the separate procurement and bidding of up and down reserves, ii) a competitive secondary energy market, iii) multiple time units instead of hourly market periods of 24 hours, and iv) a minimum bid size of 0.1 MW or MWh. Furthermore, users may select if they want to simulate coupled balancing markets from different market zones. To compute the procurement of secondary capacity, it is necessary to put either the maximum expected consumption per period under simulation or the capacity needs, as preferred by users, in the SecondaryNeeds.xls file. In the case of putting the maximum expected consumption in the file, users may select the symmetrical capacity procurement as suggested by ENTSO-E or the asymmetrical procurement as used by the Portuguese TSO. Furthermore, the TSO agent submits to the market the up and down needs of the secondary capacity and collects the agents' bids to the market (Secondary.xls file), computing the market-clearing price(s) and up and down capacities and exporting them to the file SecondaryOutput.xls.

To clear the secondary energy market, the TSO agent reads the secondary dispatch needs from the file SecondaryENeeds.xls, collects the agents' bids (SecondaryE.xls file), computes the up and down energy prices, and exports the results to the SecondaryEOutput.xls file.

To clear the tertiary energy market, the TSO agent reads the tertiary dispatch needs from the TertiaryNeeds.xls file, collects the agents' bids (Tertiary.xls file), and computes the up and down energy prices, and exports the results to TertuaryOutput.xls file. Furthermore, the TSO agent computes all costs with the reserve energy markets and the respective penalties of the imbalanced agents. Also, the TSO agent reads the day-ahead prices computed in MASCEM from a JSON file and computes the up and down imbalance prices. Those penalties are computed using the Nord Pool formulation, as adapted throughout Europe, or the Portuguese formulation, as selected by users. The penalties and imbalance prices are exported to the TertiaryOutput.xls file.

Finally, the **the-end** tool concludes the simulation removing all temporary auxiliary data project's directory.

3. TradeRES shared datasets

A common data model to serve most features of all the modelling tools was designed and a shared MySQL database server was set up to serve the modelling activities. However, finalising the common data model and getting the models to connect to the database proved to be very challenging, as described in Deliverable 2.3 (Kiviluoma et al., 2022). Three reasons were identified: 1) developing a shared ontology with shared vocabulary was impractical, 2) working from a common database is an extra burden for the modellers, and 3) the data transformations required to enable running models from a common data format are formidable.

TradeRES WP2 has compiled datasets for inputs and outputs of the optimisation models. To ensure that all models use consistent TradeRES data, several activities have been initiated and improved: creating data summary tables, automating data checks and logging data changes.

4. Discussion

In this project, we have applied the Spine Toolbox to link and co-simulate energy models with distinct scopes, achieving simulations that represent a broader market. Instead of independently expanding the modelling capabilities of each model, linking models under a common simulation platform offered the possibility of combining strengths of individual models. Nevertheless, the process of preparing the inputs and outputs and synchronizing the data was a very laborious task that consumed a significant amount of time.

The challenges of creating a unified database for the purposes of a project like TradeRES are now informing the work ongoing in the EU project Mopo (a follow-up of the EU project Spine) which is further developing Spine Toolbox to build a generic energy modelling data format, offering a reasonable data structure for all data while still leaving room for model specific peculiarities.

In the future, projects following TradeRES will not need to design (very) time-consuming bespoke data structures to exchange data between models. The work performed in TradeRES also informed the further development of data transformers in Spine Toolbox as well as the available application programming interface (API) functions, which have been recently rewritten to be more straightforward to use.

5. Conclusion

Individual modelling tools have been integrated with Spine Toolbox, and combined workflows for the case studies have been built. Spine Toolbox was chosen by TradeRES consortium as the platform for integrating the modelling tools due to its strong workflow support and database-centred architecture.

The TradeRES repositories (under https://github.com/TradeRES) include the Spine Toolbox workflows both for the individual modelling tools and for the coupling of the tools. Partly because of challenges in creating a shared ontology and the required data transformations for each tool, all workflows have their own databases instead of the planned common data model and shared MySQL database server. The challenges of creating a unified database are now informing the work ongoing in the EU project Mopo, a follow-up of the EU project Spine.

The workflows are being used in the TradeRES case studies. Integration of some modelling tools is still in progress and will be finalised alongside the case study simulations.

References

Cvetkovic, Milos, Juha Kiviluoma, Erkka Rinne, Ingrid Sanchez Jimenez and Christoph Schimeczek. (2020). *D4.3.2 – Principles and usage of a multi-simulation electricity market tool (D4.7)*. TradeRES project deliverable. https://traderes.eu/wp-content/uploads/2021/04/D4.7_TradeRES_Principles_H2020.pdf.

Deissenroth, M., M. Klein, K. Nienhaus and M. Reeg, 'Assessing the Plurality of Actors and Policy Interactions: Agent-Based Modelling of Renewable Energy Market Integration,' Complexity, vol. 2017, pp. 1–24, 2017, DOI:10.1155/2017/7494313.

Hommes, Jim (2021). Exploring Soft-linking of Market Modules and an Optimization Model. http://resolver.tudelft.nl/uuid:63691862-9a26-4df3-b6dd-57bed6c9d8a5.

Kiviluoma, Juha, Ni Wang and Silke Johanndeiter. (2022). 'How to use TradeRES optimization models database' (D2.3). TradeRES project deliverable. https://traderes.eu/wp-content/uploads/2024/01/D2.3_TradeRES_D2.3_How-to-use-TradeRES-optimization-models-database.pdf

RESTrade User Guide. https://traderes.eu/wp-content/uploads/2022/06/6-TradeRES-User-Guide-RESTrade_LNEG_vf2.pdf

Santos, Gabriel, Tiago Pinto, Isabel Praça, Zita Vale. (2016). "MASCEM: Optimizing the performance of a multi-agent system", Energy, vol. 111, pp. 513-524. DOI: 10.1016/j.energy.2016.05.127.