ENHANCED VIRTUAL POWER PLANT DESIGN AND IMPLEMENTATION LESSONS

Gary HOWORTH
University of Strathclyde – UK
gary.howorth@strath.ac.uk

Ivana KOCKAR
University of Strathclyde – UK
ivana.kockar@strath.ac.uk

Paul TUOHY
University of Strathclyde – UK
paul.tuohy@strath.ac.uk

Graeme FLETT
University of Strathclyde – UK
graeme.flett@strath.ac.uk

John BINGHAM
ETC Ltd. – UK
j.bingham@e-t-c.co.uk

ABSTRACT

Aggregation by Virtual Power Plants to provide flexibility to distribution and transmission networks is seen as an important element in the transition to Net-zero. This paper presents work carried out in the SIES 2022 ERA-Net project, which is investigating in detail the possible provision of flexibility by different technologies but thorough a lens of different business models. Thus, presented work relies on the real use cases. The focus of this work is on the overall architecture (hardware and software) of the demonstrator plant in East Kilbride Scotland and the integration of assets. The paper highlights challenges and lessons learned, during this process.

INTRODUCTION

Aggregation by Virtual Power Plants (VPP) to provide flexibility to distribution and transmission networks is seen as an important element in the transition to Net-zero. This paper presents work carried out in the SIES 2022 ERA-Net project, which is investigating in detail the possible provision of flexibility by different technologies, including EVs. Thus, presented work relies on the real use cases. One of the partners in the SIES 2022 ERA-Net consortium is the Engineering Technology Centre in Central Scotland (ETC), which has been set up to deliver a technology demonstrator system to manage energy pools using VPP software as well as to investigate how this VPP could operate using a variety of assets in a realistic setting. ETC has interests in two energy pools, which are available for immediate deployment in the project:

- ETC's own premises and the wider Scottish Enterprise Technology Park energy infrastructure
- A test area at the Myres Hill wind turbine site.

The project also has access to data from the Findhorn Eco Village in Moray, Scotland which provides data from 3 wind turbines, a shared thermal store heat pump, domestic heat and hot water usage in several houses. The ETC site includes both electrical and thermal loads that can be used for flexibility as well as from other neighbouring buildings. The VPP design includes the use of cloud-based third-party software for data storage, communication between assets, hardware interfaces and an additional pilot VPP software platform providing enhanced services such as optimization scheduling and forecasting amongst other things. Devices from various manufacturers have been incorporated into the demonstrator system, requiring appropriate integration procedures and settings. The integration of these components and development of the VPP software has proceeded on a learning by doing approach. The paper will outline the design, development of the VPP platform. In addition, the paper will discuss the challenges with the implementation of the various components and present results on the performance of the design in the context of supplying flexibility services to a DSO.

ETC SITE & HARDWARE

The current ETC site includes both electrical and thermal loads that can be used for flexibility as well as loads in nearby buildings, and also includes a 178kWh Delta Li-ion battery (Figure 1). The site is connected to Scottish Power's distribution network which is somewhat constrained in this area (Figure 2).

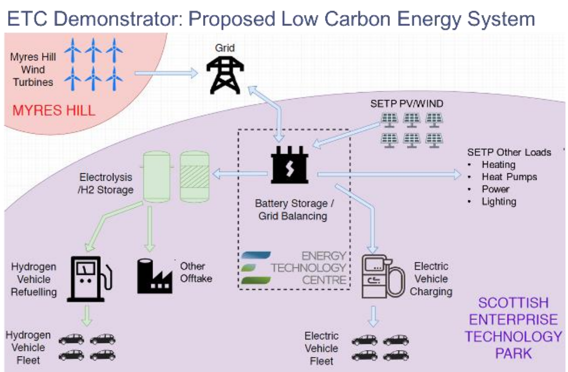

Figure 1: ETC asset overview

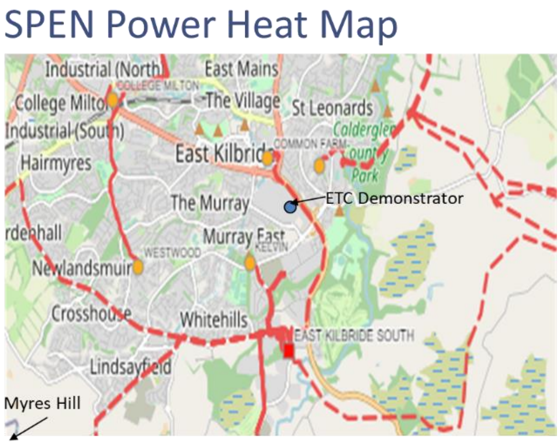

Figure 2: Distribution grid heat map

Furthermore, access to appropriate markets to generate revenues is important to a successful commercial VPP integration and implementation. The ETC site is currently too small to connect to many of these markets so the VPP software platform is able to simulate these markets with hardware in the loop. Digital twins of non-physical assets can also be added to aid in testing or trying out new combinations of assets. This approach has been used to investigate future VPP business models.

The connected distribution grid (Figure 2) has been modelled as a digital twin and provides us with a view on potential imports and exports [1]. Measurements of reactive and active power (imports and exports) on the site are being added to improve model forecasts. Assets are connected to vendors onsite PLC's via modbus connections and data is transferred to a cloud based software for storage, later analysis and asset scheduling. The VPP software is able to interact with the cloud via APIs . The VPP software currently runs on a separate server but could also be cloud based.

VPP SOFTWARE

The generic design of VPP was set out in reference [2] and consists of a number modules including communications, accounting, forecasting, scheduling, bidding and risk management etc. (Figure 3). Initially, it was envisaged that a third party supplier would provide VPP software to communicate to and from various assets and perform many of these functions. After initial scoping of the literature and available software, few solutions providers were considered, and the most promising finally selected, with the aim of building specific functions such as forecasting and optimisation in a separate prototype software environment.

However, after further extensive literature review and more detailed discussions, it become clear that none of the current software providers were able to meet all the needs of the SIES project. Although the systems provided excellent functionality in some aspects notably communications, storage and the hardware to connect to assets, many of the modules mentioned in reference [2] were missing. In addition, there were various supply issues (e.g. chip shortages), we well, as Covid-19 during the lifetime of the project. Therefore, a decision was made to utilise existing University of Strathclyde software to build a prototype VPP software environment both to communicate with field assets, collect data, send control signals, and carry out day-to-day and half hourly functions.

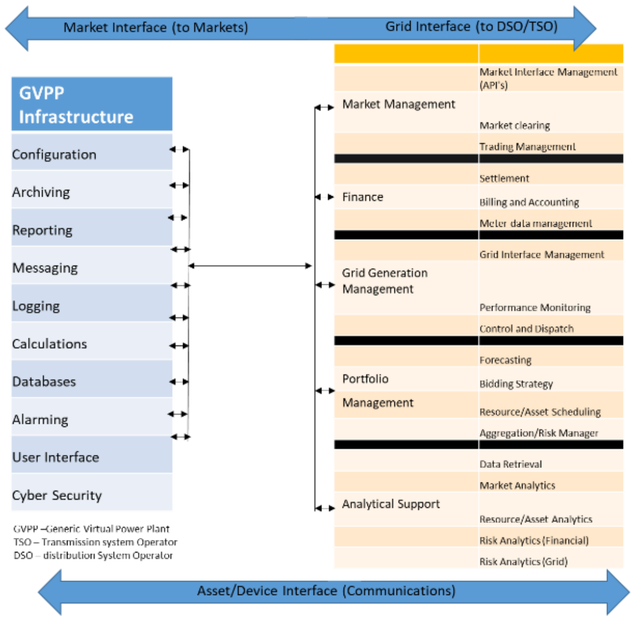

Figure 3: Generic VPP design: Adapted from [2]

PyEMLab-AGG: An Existing Framework

PyEMLab-AGG [3] was developed as a python object orientated simulator to model the interactions of aggregators (VPP owners and associated actions), domestic and industrial customers in a future flexibility market. It is a structured, ontology driven, environment that uses python as a scripting language to set up scenarios and assign roles (i.e. what to do, when and how, as well as the rules to make decisions). It is based on a python port of the java based EMLAb program [4] which was originally designed to simulate investment and technology behaviour in an European power market. Both EMLab and PyEMLab-AGG have been extensively tested.

The object-orientated program is organized into packages as summarized in Figure 4.

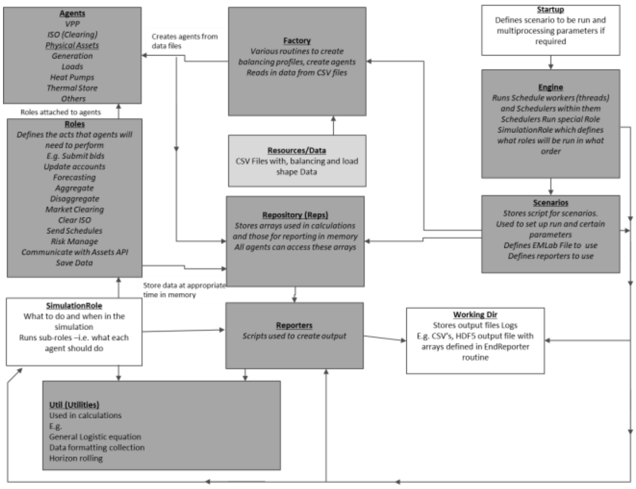

Figure 4: PyEMLab-AGG structure

Simulation runs are initiated by running a "startup" module stored in the engine package, which defines the scenario to be run e.g. sets up assets in the VPP using the factory pattern, and reads in data as required. The "startup and scenario files also define which reporters to use; reporters are used for reporting output data either during the simulation or at the end. Finally, "startup" starts a scheduler (with schedule worker threads if required for parallelisation).

CSV and other data files are stored in the resources directory and are used to populate digital twins and provide other data as required.

The utility package contains code segments that are used within the various calculations, and sub-roles. Typical routines stored in here include risk management calculations, logistic estimation routines and so on.

During program operation, data is stored in a repository that stores in-memory matrices of simulation output or data required for calculations e.g. price forecasts, near term wind measurements and so on. Results and other data can be stored as files in a number of formats including SQLite, PostgreSQL and Hdf5. Although the original PyEMLab-AGG was designed as simulator, the latest version utilised for the VPP platform uses a python based Cron scheduler [5] that allows for real time VPP operation. Finally, the roles package is used to define the actions that agents (VPP, forecasting modules) perform during the simulation.

VPP Software Architecture

Using the PyEMLab-AGG structure discussed above, the framework has been rebuilt to communicate in real time every half hour with assets at a number of locations including the ETC site.

This was performed in a step wise fashion adding and testing functionality using use cases as a guide. The first use case was based on one asset with a simple control rule. Later use cases involve additional assets such as heat pumps/thermal stores and the interaction with more complicated market structures. The current architecture for the software framework is shown in Figure 5. Those modules marked with an asterix* are for future development. A sequence diagram (Figure 6)
shows the interactions between various components for a simple use case.

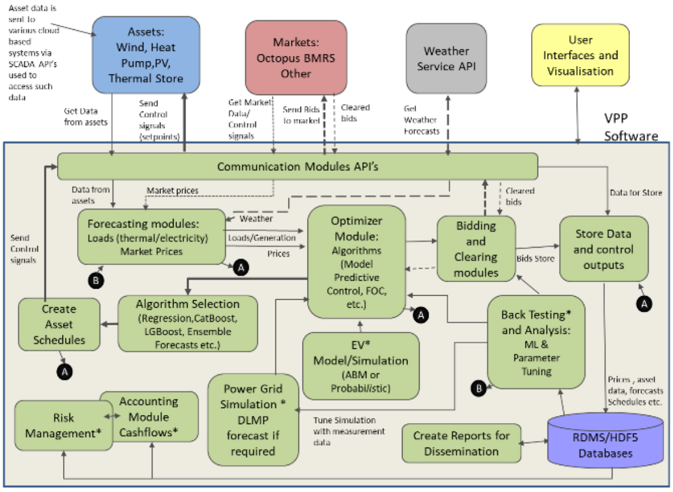

Figure 5: Architecture overview

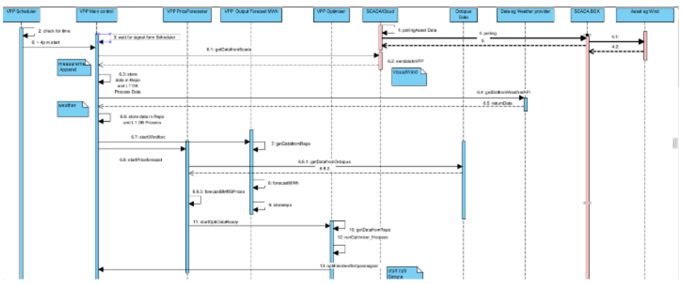

Figure 6: Sequence diagram – simple use case

A key role of the model is to communicate with the assets in the field. This is achieved using API's, some of which have had to be developed for this project. Data is collected and stored for later use, but those that are needed for immediate use are also stored into in-memory storage in the repository object described earlier.

PDF style reports are created weekly and can be emailed to the appropriate parties. The software uses a rolling time horizon to forecast prices and demand, and is used in the decision module.

A decision optimisation model, based on the work in reference [6, 7], is included, but has been adapted to include battery degradation costs import/export limits and carbon prices. It uses the Pyomo [8] optimisation model. This has been further adapted to include some elements of the current UK flexibility auction market. The optimiser, or decision model, looks to maximise revenues to the project and formulates schedules, which are then sent via the communication module to the various assets. The current model uses Octopus market prices so assumes the

VPP is a price taker. This means that that bidding module is not currently used but would be as other markets are accessed. The decision module also includes other algorithms such as load following or other simple heuristics such as buy-low-sell-high and so on. Additional algorithms can be included. Note that analysis of algorithms and data is currently carried out offline.

Data Driven Analysis

A key part of this project has been the collection and storage of data for later analysis. This allows for a data driven approach to improve forecast model building, improve their accuracy and for creating digital twins.

Hardware/Software in the Loop

The original framework was designed as simulation software but now has dual functionality. This has proved invaluable for a number of reasons. Firstly, some of the assets have not been available to the project so the use of digital twins allows us to test software functionality and to try out new algorithms. Secondly, the current demonstrator cannot gain access to some of the markets so a simulation environment allows to simulate these markets in the presence of real assets and their data.

Forecasting and Optimisation Functionality

An important component of this VPP platform is associated with forecasting (Figure 7). The algorithms used are currently deterministic but stochastic algorithms and approaches are being investigated.

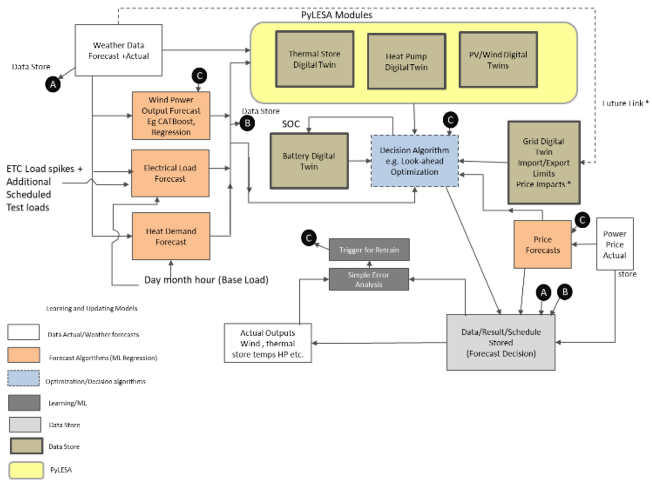

Figure 7: VPP forecasting and optimization logic

A number of forecasting modules have been constructed, some of which use existing machine learning libraries such as CatBoost [9], others that use standard regression techniques. Ensemble learning [10] is to be included in a future version.

In addition, PyLESA [11] is an open source modelling tool for the design of local, integrated and smart energy systems and includes calculations and modules for solar/wind assets, heat pumps and thermal stores (digital twins). It also uses the Gekko [12] optimisation model to perform forward looking model predictive and fixed order control of the assets. Various components of PyLESA have been integrated into the modified PyEMLab framework and are to be used to optimise heat pump and thermal store assets.

Digital Twins

A number of digital twins have been constructed and are represented as agents within the software framework. In particular a distribution grid model has been constructed that will allow for better forecasting of future grid constraints [1], as well as a battery model to mimic the Delta battery.

CHALLENGES

Hardware Challenges

ETC consists of a number of different types of assets from many different manufacturers, many of which are legacy assets. Integrating these assets into a VPP platform has proved to be time consuming process and more difficult than a simple plug and play. Defining communication channels and measurement specifications has also been challenging, especially as some of these measurements are not currently sourced.

Software Challenges

The VPP software platform has been operating for around 6 months, storing data from various locations into Hdf5 files for later use.

Although the use of an existing framework has been useful, considerable effort has been extended in adding error capture and communication retry routines to deal with issues such as with data capture, or communication errors.

Forecasting is an important element of the VPP functionality and the accuracy of those forecasts is important to the commercial success of the VPP. Note that forecasting day-ahead electricity market prices is known to be easier than those associated with the flexibility markets and those flexibility market forecasts result in relatively large errors at time. This is currently an area of active research.

LESSONS LEARNED

After an extensive literature review and more detailed discussions, it has become clear that none of the current VPP providers are able to meet all the needs of the SIES project. An off the shelf solution with cloud resources, cyber security and error capturing functionality may have proved to be a better approach, although it will still have necessitated the development and integration of specialised algorithms and routines.

Furthermore, integrating legacy assets or assets from different manufacturers has been time consuming.

However, development of VPP platform based on prior work using the PyEMLab-AGG framework has allowed us to follow a rapid prototype development process. The rapid prototyping process includes three steps; prototyping, testing, and refining (learning by doing approach). Prior experience tells us that significant software savings can be achieved with this approach as software functionality specification can be improved. Other lessons learned include:

- Choosing the appropriate level of detail in a digital twin model can be very important as shown by the example of modelling in reference [1]. Providing too much detail can just add complexity with little gain in accuracy and can result in larger computational loads.
- Uncertainty in market price and generation from renewable resources, such as wind, is relatively large. Probabilistic approaches are required, which is driving VPP owners towards using risk management techniques to protect against potential downsides.
- As the number of types of assets and their numbers grow, complexity of models increases, resulting in longer optimisation solution times. In the case where the problem is inherently non-linear, very long solution times can occur e.g. > three hours. This can be remedied by linearization and assumption simplifications, but it is important to understand how these can influence accuracy of obtained solutions.
- Pattern recognition matching could be used to help reducing solution times, or when optimisers do not converge.
- Pre-processing of data to exclude spurious scenarios helps improve forecasting, as blindly taking all data can result in worse forecasting models. This is difficult to spot in an online automated model, so human intervention/analysis is often needed.

CONCLUSIONS

A VPP platform (hardware and software) is currently being developed at the ETC site in East Kilbride. Using a learning by doing approach, existing and new assets and models are being integrated into a VPP platform. The platform is being developed using an existing software framework PyEmLab_AGG, with aim of simulating Aggregators (VPP owners). Additional elements of code, such as that from PyLESA and other available or newly developed optimization routines, have been included. There is still much work to do, but the acquisition of data has proved to be useful in that it is helping the team understand limits of constraints and influences of forecast inaccuracies when operating real assets.

Acknowledgments

The SIES 2022 project has received funding from the ERA Learn /European Union's Horizon 2020 research and innovation program under grant agreement No 811171.

REFERENCES

[1] G. Howorth, I. Kockar, P. Tuohy, and J. Bingham, "An enhanced virtual power plant for flexibilty services into a local area (including EV's)," in *CIRED Porto Workshop 2022: E-mobility and power distribution systems*, 2022, vol. 2022, pp. 970-974.

[2] S. You, C. Træholt, and B. Poulsen, "Generic Virtual Power Plants: Management of distributed energy resources under liberalized electricity market," in *8th International Conference on Advances in Power System Control, Operation and Management (APSCOM 2009)*, 2009, pp. 1-6.

[3] G. Howorth, "Extending the AgentSpring/EMLab Tool to Evaluate Additional Agent Behaviour such as Electric Vehicles and Demand Side Response," ed. ETP Annual Conference 2019 - Energy Technology Partnership Dundee UK: ETP, 2019.

[4] L. J. De Vries, É. J. L. Chappin, and J. C. Richstein, "EMLab-Generation An experimentation environment for electricity policy analysis," 2013.

[5] R. Peters, "cron," *Expert Shell Scripting,* pp. 81-85, 2009.

[6] B. L. S. Aigner, "System modeling and dispatch schedule optimization of combined PV battery system using linear optimization," Masters, University of Agder, 2021.

[7] E. Barbour and M. C. González, "Projecting battery adoption in the prosumer era," *Applied energy,* vol. 215, pp. 356-370, 2018.

[8] M. L. Bynum *et al.*, *Pyomo-optimization modeling in python*. Springer, 2021.

[9] A. V. Dorogush, V. Ershov, and A. Gulin, "CatBoost: gradient boosting with categorical features support," *arXiv preprint arXiv:1810.11363,* 2018.

[10] O. Steinki and Z. Mohammad, "Introduction to ensemble learning," *Available at SSRN 2634092,* 2015.

[11] A. Lyden, G. Flett, and P. G. Tuohy, "PyLESA: A Python modelling tool for planning-level Local, integrated, and smart Energy Systems Analysis," *SoftwareX,* vol. 14, p. 100699, 2021/06/01/ 2021.

[12] L. D. Beal, D. C. Hill, R. A. Martin, and J. D. Hedengren, "Gekko optimization suite," *Processes,* vol. 6, no. 8, p. 106, 2018.