

Development of a modelling and simulation methodology for hierarchical energy system scenarios

Johannes Mast^{a,*}, Stefan Rädle^a, Joachim Gerlach^a and Oliver Bringmann^b

^a Department of Computer Engineering, Albstadt-Sigmaringen University, 72458 Albstadt, Germany

^b Department of Embedded Systems, University of Tübingen, 72074 Tübingen, Germany

Abstract

In the context of renewable energies, precise and reliable forecasts are necessary to ensure a sustainable and economic operation of energy systems. This paper presents a methodology and environment that allows modelling, simulation and analysis of decentralized and heterogeneous energy system scenarios on a virtual level. An essential requirement for the modelling and simulation of complex network scenarios is a capable abstraction strategy, which allows hierarchical clustering and interaction of several network levels. The approach presented in this contribution provides mechanisms that allow a realistic modelling of complex energy system scenarios covering several network levels, taking into account the specific components, interactions and relationships that exist at each level (e.g. specific electrical lines etc.). The simulation follows a hierarchical tree-based approach, which allows to integrate already existing simulation models into the environment and to generate only a small overhead over the simulation run time. As we can show, the developed environment not only serves as an excellent basis for analysis and optimization purposes but also for the application of artificial intelligence techniques, as it supports the effective generation of training data and verification of optimization results.

Keywords: energy systems, energy planning, renewable energy, power grid, simulation, deep-first search

1. Introduction

In Germany, the Federal Government expects to provide 40-45 percent of the energy demand with renewable energies by 2025 [1]. The existing energy sources (nuclear energy, coal, oil, gas etc.) will be successively replaced by an energy supply based on renewable energies (wind power, solar energy, hydroelectric power, biomass, geothermal energy etc.). These far-reaching changes will arise new challenges in the planning, analysis and optimization of energy systems that are too complex for analytical or computational solutions. In such cases, simulations make it possible to check and improve system properties and behavior comparatively quickly and cost-effectively, without causing high costs or system risks. In the field of power engineering systems, they make it possible to systematically test systems under real, non-reproducible weather conditions and to repeat these tests as often as required in order to adapt and optimize system parameters and control strategies. In order to address these challenges, this paper presents a modelling and simulation methodology that allows for simulating and analyzing individual energy plants, small clusters up to complex network scenarios.

The paper is structured as follows: Section 2 describes the current state of the art of environments for modelling and simulation of energy systems. Section 3 discusses the functionality, architecture, and simulation semantics of the proposed modelling and simulation methodology. Section 4 describes the application of our methodology to a system scenario. In Section 5 the results are summarized and an outlook of future work is given.

* Manuscript received October 12, 2018; revised May 4, 2019.

Corresponding author. E-mail address: mast@hs-albsig.de.

doi: 10.12720/sgce.8.4.383-391

2. Related Work

In the field of energy technology, the increasing relevance of balanced energy networks has led to a large number of simulation programs being developed by scientific institutes. The best-known representatives for simulation environments focusing on the simulation of electrical and thermal processes are described in the following paragraphs.

Quaschnig presented an environment called *DESIRE* for the simulation of regenerative energy systems as part of its research project at the HTW Berlin. *DESIRE* is a free-available simulator that is based on Microsoft Excel as the underlying calculation concept. Components and function libraries programmed in high-level languages are connected as libraries (DLLs) via Excel macros. [2]

With *INSEL*, *Schumacher* created a modular simulation environment for the simulation of renewable energy systems, which was originally developed at the University of Oldenburg in a project funded by the German Federal Ministry of Research and Technology (BMFT) and subsequently further funded by the Volkswagen Foundation of Lower Saxony. The component library supplied with the simulation environment consists mainly of components for the simulation of solar power and solar thermal. Since version 8.2 a complete compatibility with MATLAB/Simulink is given. [3]

With *CARNOT Wemhöner et al.* presented a toolbox for MATLAB/Simulink for simulating building and heating technology, which was a research project at Aachen University of Applied Sciences. It extends the Simulink library with components such as heat pumps, heat accumulators and photovoltaics. The construction of energy scenarios is based on a low level of abstraction. [4]

In a previous edition of this journal, *Kim et al.* presented a simulation environment designed for microgrid systems, where the producers and consumers are connected in a bidirectional manner. Thereby it is possible to monitor the power of the entire power grid, the partial grid, and the single machines. [5]

When considering the existing solutions, particular attention was paid to the following criteria:

- The software should be freely available for scientific purposes.
- The software should offer a free extensibility and insight into the implementation.
- The software should be able to flexibly integrate third party simulation models and be able to perform analyses on the simulation results directly in the environment.
- The environment should provide mechanisms for hierarchical modelling, which allows to model large clusters and scenarios with different network levels.

None of the considered solutions fully meet these requirements, therefore this paper presents the development of a simulation environment that tries to meet these requirements in the best possible way.

3. Methodology

In this section, the structure of our modelling and simulation environment is explained, which allows to simulate energy scenarios, visualize results and perform analyses.

3.1. General information

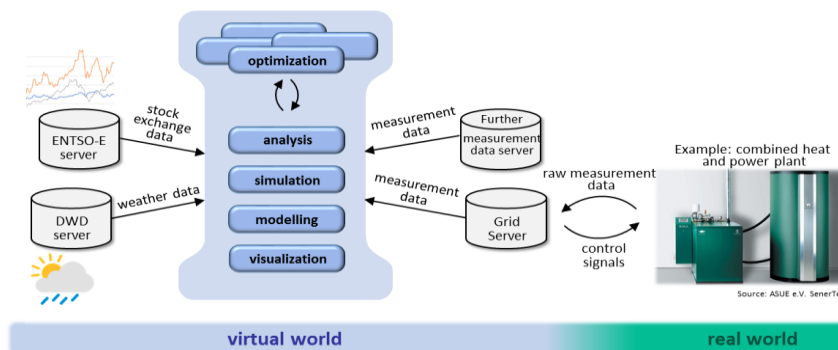


Fig. 1. Surroundings of the modelling and simulation environment

Fig. 1 shows our environment with all components that are involved. Key components of the environment provides mechanisms for the modelling, visualization, simulation and analysis of energy systems. Interfaces to data servers allow real-live measurement data, weather data and stock exchange data to be included in the simulation.

All data required for the simulation, such as weather parameters, are automatically downloaded from the connected servers for a specified time interval and geographical position. In some cases, measurement data such as heat demand is also required, which can be obtained from measurement data servers such as the *grid server*, which is operated by the partner company involved in the project.

Our environment integrates existing simulation models. It does not matter in which programming or simulation language the models were built, as long as the interfaces of the models comply with a given format. These interfaces are necessary to parameterize the model within the environment and to interpret the simulation results correctly. The simulation results can be displayed in terms of diagrams. In addition, analysis tools are provided to calculate the economic efficiency on the electricity market and the CO₂ emissions over a specified time period. Moreover our environment provides mechanisms for modelling system scenarios in a hierarchical way, which allows clusters of energy components to be combined and encapsulated into a single, more abstract component, being part of a cluster on a higher network level.

3.2. Architecture

The *Model View ViewModel (MVVM)* [6] is used as the architectural pattern for the environment. This is a special variant of the *Model-View-Controller (MVC)* pattern, whose main focus is the separation of display, logic, and data (see Fig. 2). The *View* corresponds to the visual appearance of the application. The *ViewModel* links the *Model* (datasets) with the *View* (user interface) and forms the core of the *MVVM* pattern. In contrast to the control from the *MVC* pattern, the *ViewModel* offers programming logic and the option of changing the model. In the application, the *model* represents the dataset, called the data context, for the docked *ViewModel*. In the data context, the environment stores all data that is important for successful simulation execution. Thus, the components displayed on the user interface are stored in a list in the data context. When an object is added to the list, the underlying Observer pattern automatically informs the user interface (*View*), which then displays the object. Furthermore, all time series that were obtained via a data server or imported via a file are stored in a list in the data context.

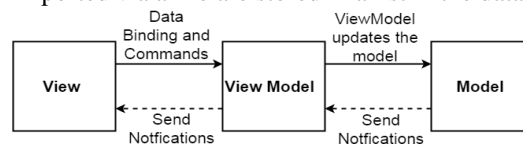


Fig. 2. MVVM-Pattern

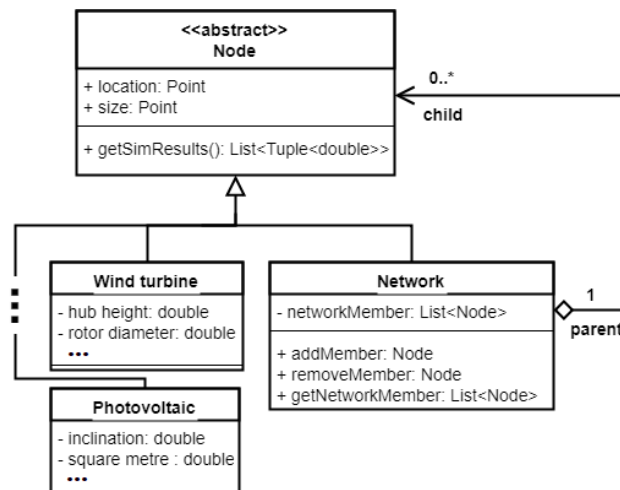


Fig. 3. Composite pattern in the application

The components that are displayed on the user interface and parameterized from there are structured according to the *composite* pattern (see Fig. 3). There is an abstract class *Node*, which bundles all properties required to be displayed as a component on the user interface. All power plants such as wind turbines and photovoltaic units are derived from *Node* and add their specific characteristics. In addition, there is a network component for the hierarchical structure of scenarios that can accommodate any number of nodes.

3.3. Simulation

Within the environment, graphs are created in which energy components are represented as nodes and connecting lines as edges. All simulable graphs $G(V, E)$ are root trees. By definition, these are connected graphs, without circles, which have exactly $|V| - 1$ edges and have exactly one path between each two nodes.

Already in the 19th century, *Gustav Kirchhoff* analyzed electric circuits in terms of tree graphs and established important connections of its well-known Kirchhoff equations with the number of spanning trees of the switching network [7].

The simulation uses external simulation models that comply with predefined interfaces so that they can be executed from the environment with the parameters specified there. The root must always be a meter. From there, a depth-first search starts, which traverses the nodes from top to bottom. However, the models are not executed until the backtracking phase has started.

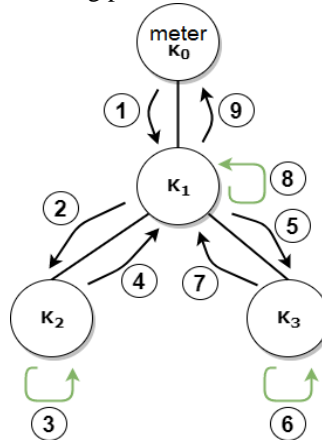


Fig. 4. Simulation semantics

Fig. 4 shows the process for a root tree with three components and a meter. In the first two steps, the counter runs through the nodes until a leaf node is found. In the third step this node is simulated. Each node generates a time series during its execution, where $\mathbf{x} \in \mathbb{N}^n$ represents the time points and $\mathbf{y} \in \mathbb{R}^n$ the corresponding values, in which n is the number of discrete time steps described.

Depending on the distance and wiring of the nodes, a transmission loss is calculated for the individual power values in the time series. First, the generated power is divided by the voltage of the power grid.

$$I_{cable} = \frac{P_{gen}}{U_{grid}} \quad (1)$$

The voltage varies strongly depending on the network level: local distribution networks are usually operated with a voltage of 1kV, while transmission networks rely on voltages of up to 380 kV. In addition to the voltage, the resistance of the power cable is taken into account. This results from the material constant ρ , which is $0.017\Omega \frac{mm^2}{m}$ for copper, as well as the length and cross-sectional area of the cable. It should be noted that the cable resistance must be calculated for the outward and return path.

$$R_{cable} = \rho * \frac{l}{A} \quad (2)$$

Subsequently, the transmission loss can be calculated.

$$P_{loss} = I_{cable}^2 R_{cable} \quad (3)$$

When the simulation results are passed on to the predecessor nodes, the transmission loss of each value of the time series \mathbf{y} is subtracted from for the edge $e = \{K_{l-1}, K_l\}$. It applies:

For all y_i in \mathbf{y} :

$$y'_i = y_i - \left(\left(\frac{y_i}{U_{grid}} \right)^2 * \rho * \frac{l}{A} \right) \quad (4)$$

This calculation can be computed much more efficiently on the computer if the transmission loss is not calculated individually for each power value. Therefore P_{loss} is transformed to:

$$\begin{aligned} P_{loss} &= \left(\frac{P_{gen}}{U_{grid}} \right)^2 * \rho * \frac{l}{A} \\ &= P_{gen}^2 * \left(\frac{1}{U_{grid}^2} \right) * \rho * \frac{l}{A} \end{aligned} \quad (5)$$

In this form, the element by element multiplication of both vectors can be used to create a linear mapping $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$, that transforms a vector $\mathbf{y} \in \mathbb{R}^n$ to a vector $\mathbf{y}' = F(\mathbf{x}) \in \mathbb{R}^n$.

$$\mathbf{y}' = \mathbf{y} - (\mathbf{y} \circ \mathbf{y}) \left(\left(\frac{1}{U_{grid}} \right)^2 * \rho * \frac{l}{A} \right) \quad (6)$$

Before K_1 can be simulated, K_3 must be simulated (step 6), and pass its results to K_1 (step 7). The processed simulation results $\mathbf{y}'_2 \in \mathbb{R}^n$ and $\mathbf{y}'_3 \in \mathbb{R}^n$ from nodes K_2 and K_3 are summed up after the seventh step to a vector \mathbf{y}_1 and passed to the immediate predecessor node K_1 . When m is the number of nodes on the lower level, it generally applies:

$$\mathbf{y}_{l-1} = \sum_{k=l}^m \mathbf{y}'_k \quad (7)$$

In the eighth step, K_1 itself is simulated and the results are combined with \mathbf{y}_1 . After the ninth step, the root K_0 has all results of the children (here only K_1) and can unite them so that the highest counter has the total result for the built scenario.

If the node to be executed is a network component, the procedure is the same as for the main tree: In the network, the root tree is traversed from the meter according to the depth-first search until results are available in the meter. These are passed out of the network. The next node in the simulation sequence is then continued as usual. If the simulated scenario contains wind power or photovoltaic plants, the weather data for the selected period is downloaded from the weather data server and included in the simulation.

A simulation is successfully completed when the backtracking process has arrived at the original root node and all subjacent nodes in the tree have propagated their simulation results up to the root. This simulation semantics is possible when the models of the component library are not interdependent. If such interdependencies exist between devices, they are covered within the external simulation model. For example, the dependencies of an internal combustion engine with a buffer storage are determined for a combined heat and power unit (CHP) is taken into account within the external simulation model. In contrast to event-driven simulation, the runtime is not reduced by context changes, which increase

exponentially in terms of the number of participants. Also the programming or simulation language in which the models were created is not important as long as the model interfaces use the same formats.

3.4. Analysis tools

Within the environment various analysis tasks can be performed on the results of individual plants or entire clusters. The analysis tasks are encapsulated in a modular manner which flexibly allows to add new analysis mechanisms to the environment.

(1) Calculating the volume in the electricity grid

The volume provides an overview of the quantity of consumption or generation over a defined period of time. This information can help in planning enhancements, for example, by revealing the relationship between generation and consumption. The time series are also required for many other calculations.

For the time series of the simulation, which are expressed by the vectors $\mathbf{x} \in \mathbb{N}^n$ with the corresponding $\mathbf{y} \in \mathbb{R}^n$ the vector $\mathbf{a} \in \mathbb{R}^n$ can be calculated with:

For all a_i in \mathbf{a} :

$$a_i = \begin{cases} 0, & i = 1 \\ a_{i-1} + \frac{1}{2} * (y_i + y_{i-1}) * (x_i - x_{i-1}), & i > 1 \end{cases} \quad (8)$$

To calculate the area between two points in time, the formula for the area of a trapezoid is applied. In addition, the former area is added to the result to create an accumulated time series that contains the total value in the last element (e.g. total generation in watt-hours (Wh) for one day if it is a time series of a plant over one day).

(2) Calculating the earnings in the electricity grid

The profit of a cluster can be determined by using the electricity prices on the stock market of the electricity exchange (here *EPEX SPOT*). This enables the environment to check and optimize various control strategies for days in terms of profit. Via the data interface to the *ENTSO-E* server, the prices for the simulated period are obtained from the European electricity exchange *EPEX SPOT* (either intraday or day-ahead market). This time series for the prices is denoted by the vectors $\mathbf{x} \in \mathbb{N}^n$ and $\mathbf{p} \in \mathbb{R}^n$ (assuming that the time series have the same discrete time points). Since the prices on the electricity exchange are only valid for one hour at a time, the time series contains a price for every full hour. The prices are multiplied with the amount of energy that can be determined from the volume time series (see subsection (1)). If there are no values for the full hours in the volume time series, these must be interpolated. The yields $\mathbf{v} \in \mathbb{R}^n$ can be calculated with:

For all v_i in \mathbf{v} :

$$v_i = \begin{cases} 0, & i = 1 \\ p_{i-1} * (a_i - a_{i-1}), & i > 1 \end{cases} \quad (9)$$

The total profit P over the period of \mathbf{v} as a scalar value is calculated as follows:

$$P = \sum_{i=0}^n v_i \quad (10)$$

(3) Calculation of CO_2 emissions in the cluster:

The analysis of the CO_2 emissions can be used to draw up balance sheets and, for example, to plan whether additional CO_2 licenses need to be acquired. Depending on the fuel selected, the CO_2 emissions of a combustion engine change. If a combined heat and power unit is operated with natural gas, it emits 0.20 kg/kWh CO_2 . This is less than with other fuels such as heating oil. [8]

The volume curve can be used to calculate how much energy a system has generated over a period of time. If this is a fossil fuel plant, the CO_2 emissions generated can be calculated over the period, e.g. for a natural gas CHP unit:

$$c_{\text{CHP}} = \frac{a_n}{1000} * 0,20 \quad (11)$$

Renewable energies generally do not generate any CO₂ emissions during operation.

4. Validation

In the following, the developed modelling and simulation environment is applied to a hierarchical energy system scenario with two grid levels. The scenario is located at the geographical position (48°N, 9°E), which is in the area of Stuttgart, Germany. For the time period of one year (ranging from 01.01.2016 to 31.12.2016), the corresponding real-life weather conditions for that area and the real-life electricity costs at the electricity stock exchange are taken into account.

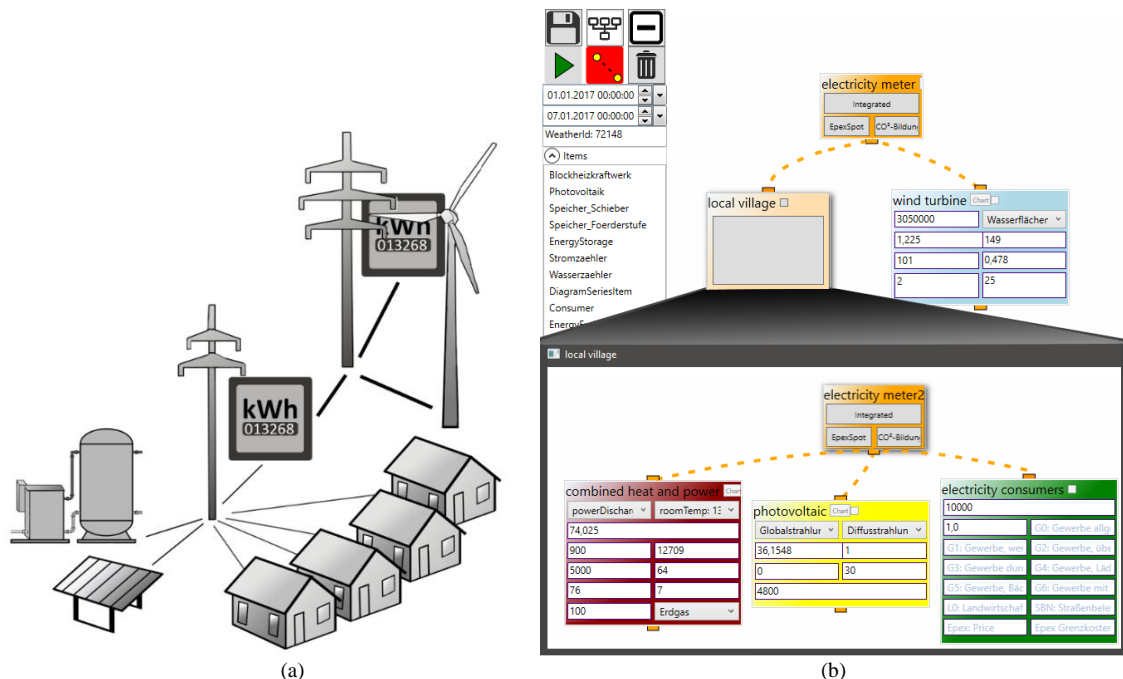


Fig. 5. Scenario with two network levels: (a) sketch (b) transfer to the modelling and simulation environment.

4.1. Construction of a hierarchical virtual system scenario

Fig. 5a shows a scenario with two network levels. At the upper level, electricity is distributed via a wider cross-section of electrical lines, operating on a higher voltage level, which reduces the transmission loss. At this level, turbines with a higher output typically feed into the grid - in this case, a wind turbine. At the lower grid level, consumers – in this case private households - receive their electrical power on a low voltage level and typically smaller producers such as Micro-CHP units and small photovoltaic systems feed their generated electricity into the grid.

Fig. 5b shows the transfer of the scenario into a hierarchical virtual system model within our environment. The main window shows the top grid level, consisting of a component which models the wind turbine, specified by parameters such as hub height, rotor diameter and roughness class, and a network component which contains the lower grid level. The lower hierarchy level can be edited and refined in the same manner within another window. At that lower grid level, three components are used to represent a CHP unit, a photovoltaic system and a consumer. The CHP unit is specified by parameters such as nominal output, storage capacity and temperature limits, and the photovoltaic system by parameters such as square meters, inclination and orientation. The consumer component is able to generate demand curves that map several consumers (in our case several private households). The cable properties such as distance, voltage and cross-section can be entered via the edges. As explained in section 3.3, the transmission losses are continuously subtracted from the transferred power values.

4.2. Simulation and analysis of the virtual system model

When simulating the scenario with the settings of Fig. 5b, Fig. 6a shows the volume curve (calculated according to section 3.4) of the lower grid level for the considered time period. It shows that total production and total consumption during that year are fairly similar, since at the end of the year the accumulated value is close to zero for both variants. However, in the winter months, production is significantly higher than consumption and vice versa in the summer months. This is due to the CHP unit that depends on the heat demand of the consumers, which is much higher in the winter months. Although there is still a photovoltaic system on the lower level that generates the largest part of its electricity in the summer months (when the irradiance is high), it is relatively small compared to the CHP unit. The photovoltaic system has a watt peak of $4800 W_p$ and the CHP unit has a nominal output of $5000 W_{el}$. Moreover, the CHP unit reaches significantly more running time with nominal output. A large variance in generation and consumption over time leads to a large regulation effort. Whenever the scenario generates too much electricity, it needs to be traded (e.g. on the European electricity exchange). Similarly, if there is a greater demand for electricity, the missing electricity needs to be purchased externally. This leads to additional regulation effort.

In the considered scenario, a more suitable matching in the lower grid level can be achieved by making the CHP unit smaller and the photovoltaic system larger in terms of nominal output.

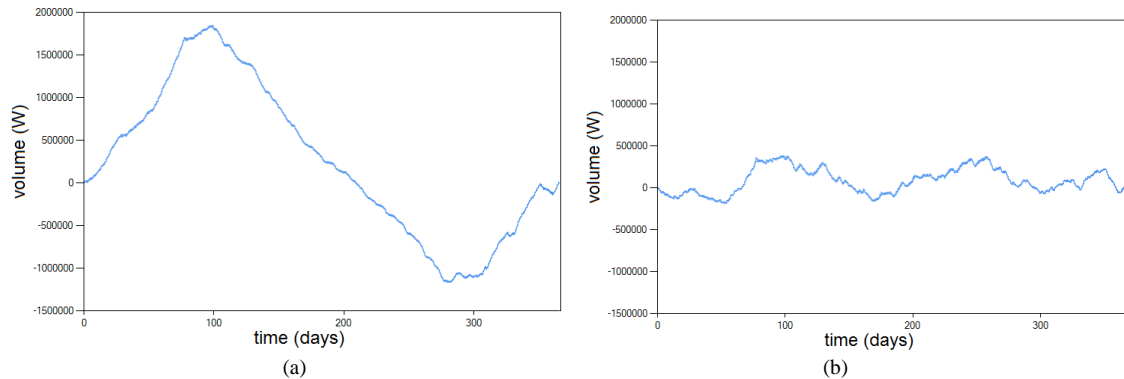


Fig. 6. Simulated Volume of Scenario: (a) inappropriate dimensioning (b) appropriate dimensioning

For the scenario with the same consumers, Fig. 6b shows the volume curve when the nominal output of the CHP unit is reduced to $3800 W_{el}$ and those of the photovoltaic system is increased to $9400 W_p$. As the characteristic curve shows, the seasonal effects may be compensated more effectively taking care of the specific characteristics of the technologies used.

When the regulation of the energy for the two different dimensioned scenarios is done on the electricity exchange, the resulting costs of both approaches are quite similar, but for the appropriate dimensioned variant (Fig. 6b) the total volume of trading capacity can be significantly reduced compared to the inappropriate dimensioned variant (Fig. 6a), corresponding to a minimization of administrative costs. So when targeting on the self-sufficiency of the regarded cluster, the variant in Fig. 6b matches the target criteria much more efficient.

The experiment demonstrates that a considerable impact can be achieved by an appropriate dimensioning of the technologies, which means that less effort has to be invested in the regulation of energy flows.

5. Conclusion

Within this paper, our modelling and simulation methodology and environment was presented, which provides a fast and highly accurate analysis of complex energy system scenarios in terms of several, also combined target criteria on a virtual base and in a hierarchical fashion. This allows for systematically

answering questions and exploring alternatives on the level of individual plants, cluster, up to large energy networks. An application of the approach to a virtual scenario demonstrates the handling and capabilities of the environment.

Future work will explore opportunities to use the approach as test and data generation environment in the context of artificial intelligence techniques to be applied to the “intelligent” optimization of energy system scenarios.

Reference

- [1] BMWi. (January 2017). Die Energiewende: unsere Erfolgsgeschichte. [Online]. Available: <https://www.bmwi.de/Redaktion/DE/Publikationen/Energie/energiewende-beileger.pdf>
- [2] Quaschnig V. Dynamisch erweiterbare Simulation von Photovoltaiksystemen mit DESIRE. In: Tagungsband 23. Symposium Photovoltaische Solarenergie, 2008:152-157.
- [3] Schumacher J. (March 2014). INSEL 8 Tutorial. [Online]. Available: http://www.insel.eu/files/public/inselTutorial_en.pdf
- [4] Wemhöner C, B. Hafner und Schwarzer, Simulation of solar thermal systems with CARNOT blockset in the environment Matlab Simulink,“ in Proc Eurosun 2000 conf, 2000.
- [5] Kim HH, Tak HS, Cho HG, Kim HJ. Development of simulator for microgrid system. *International Journal of Smart Grid and Clean Energy*, 2018; 7(3): 195-201.
- [6] Sossella V. (April 2018). Exciting secrets about MVVM that nobody tells you. [Online]. Available: <https://android.jlelse.eu/exciting-secrets-about-mvvm-that-nobody-tells-you-a95548ea684b>
- [7] Kirby E, Mallion R und P. Pollak. (January 2017). What Kirchhoff Actually did Concerning Spanning. In: Croat. Chem. Acta, 2016.
- [8] Quaschnig V. Regenerative Energiesysteme. 9rd ed. München: Carl Hanser Verlag; 2015.