The Open Energy Modelling Framework (oemof) - A novel approach in energy system modelling

S. Hilpert[a,b,*], C. Kaldemeyer[a,c], U. Krien[d], S. Günther[a,b], C. Wingenbach[a,b], G. Plessmann[d]

[a]*Center for Sustainable Energy Systems (ZNES), Flensburg*
[b]*Department of Energy and Environmental Management, Europa-Universität Flensburg, Auf dem Campus 1, 24943 Flensburg, Germany*
[c]*Department of Energy and Biotechnology, University of Applied Sciences Flensburg, Kanzleistraße 91-93, 24943 Flensburg, Germany*
[d]*Reiner Lemoine Institut gGmbH, Rudower Chaussee 12, 12489 Berlin, Germany*

Abstract

Energy system models have become indispensable to shape future energy systems by providing insights into different trajectories. However, sustainable systems with high shares of renewable energy are characterised by growing cross-sectoral interdependencies and decentralised structures. To capture important properties of increasingly complex energy systems, sophisticated and flexible modelling environments are needed. This paper presents the Open Energy Modelling Framework (oemof) as a novel approach in energy system modelling, representation and analysis. The framework forms a structured set of tools and sub-frameworks to construct comprehensive energy system models and has been published open source under a free licence. Using a collaborative development approach and extensive documentation on different levels, the framework seeks for a maximum level of transparency. Based on a generic graph based description of energy systems it is well suited to flexibly model complex cross-sectoral systems ranging from a distributed or urban to a transnational scale. This makes the framework a multi-purpose modelling environment for strategic planning of future energy systems.

Keywords: decision support, energy system modelling, optimization,

---

[*]Corresponding author, Phone: +49 (0) 461 805 3067
Email address: `simon.hilpert@uni-flensburg.de` (S. Hilpert)

collaborative development, open science

1. Introduction

The global transition process towards more sustainable and low-carbon energy systems requires the development of alternative future trajectories for a profound scientific discussion. Based on this, decision processes on different levels e.g. in transnational policy making or local energy planning can be supported. However, future energy systems imply a rising complexity in technical, economical and socioeconomic dimension due to increasingly cross-sectoral and decentralised structures [1]. Insights into such complex systems can be gained by applying computer based modelling approaches which create a quantitative basis for the abovementioned discussion and decision processes [2].

Depending on the specific investigation and research question, a variety of model types can be applied. Among others, such model types include power flow models for electricity transmission network operation and planning, economic dispatch models for general capacity planning and unit commitment models for power plant utilization [3, 4, 5, 6]. Applications range from large scale transnational investigations using purely economical top-down equilibrium models to detailed technical local infrastructure planning using bottom-up models based on technology-specific measurements. Moreover, many models can be adapted to integrate different sectors such as electricity, heat and mobility to investigate cross-sectoral interdependencies.

This paper presents the Open Energy Modelling Framework (oemof) as a new approach in energy system modelling. Firstly, the idea of a single energy modelling framework is differentiated from other approaches such as specific models or model generators in Section 2. Secondly, the philosophy, the underlying concept with its mathematical representation and the framework implementation are outlined in Section 3. Building on this, the general process of application development is described and existing applications are presented in Section 4. Finally, the scientific contribution is summarized in Section 5.

2. Status quo

Energy system models and derived results have often been heavily discussed among different stakeholders and being criticized for not opening their internal logic and underlying assumptions [7, 8]. As a result, during the last decade more scientists have opened their models and data [9]. This improves the overall transparency, enables reproducibility and allows other people to re-use or adjust tools that have already been developed. Thus, a rough division into a line of closed (*1st generation*) and open (*2nd generation*) models for energy system analysis can be derived in which these can be further classified by their respective purpose. As the European Commission has recently started to push open science in their research programmes [10] , the subject of openness has finally moved into the public spotlight which gives a forceful support to open modelling approaches. This section is designed to provide a brief overview on relevant energy system modelling software of the *1st* and *2nd generation*. For extensive reviews on energy system modelling it is referred to Hall and Buckley [11] and Pfenninger et al. [1].

2.1. Modelling landscape

In the following, it is distinguished between the three terms *model*, *model generator* and *framework*. Models are concrete representations of real word systems (e.g. with a specific regional focus and temporal resolution). Such a representation may consist of multiple hard- or soft-linked sub-models to answer clearly defined research questions. Models can be build from model generators that allow to build models with a certain analytical and mathematical approach (e.g. by the use of predefined set of equations, represented technologies). Note, that this definition is similar to the concept of abstract (mathematical description, no data) and concrete (an instance of an abstract model populated with data) models in mathematical programming. Finally, a framework can be understood as a structured toolbox including sub-frameworks and model generators as well as specific models (e.g. wind feed-in models).

The *1st generation* models / model generators have a long tradition and are predominant in the academic energy system modelling landscape. Among the most widely used proprietary model generators are MARKAL/TIMES [12]. Based on this, many models of a specific family have been build to answer research questions in the field of energy planning which is indicated by the high number of references in academic literature [11]. Similarly, MESSAGE is a prominent example and used for example for the IIASA global energy scenarios [13]. The partial equilibrium simulation model POLES [14] is another tool that is used for long term planning and forecasting. One application example is the simulation of EU-28 energy system including complete energy processes from production to final consumption. Besides this, the EnergyPlan simulation model has been applied in various research projects to analyse sector integrated electricity, heat and transport systems [15].

The Balmorel model [16] is one of the first *2nd generation* energy system models. It can be used for power and heat dispatch modelling with optional investment within the Baltic region and is written in GAMS. Another early project is the model generator OSeMOSYS [17] which is mainly used for long-term integrated assessment and energy planning. This project aims to facilitate modelling and education due to a free software approach and a fast learning curve of the software. Since then, various other projects with different purposes have been developed. A subset of them is listed in Table 1.

Their focus covers the full range from power flow simulation to long-term investment models. More open source models can be found on the website of the Openmod-Initiative [9]. While some of these projects are models for a specific region (e.g. EMMA for Western-Europe), others can be classified as model generators.

2.2. Delimitation

As the source code of *1st generation* models is not accessible, an adequate comparison between these projects and our approach is not possible. Therefore, this analysis is restricted to approaches of the *2nd generation* of models.

Table 1: **Selected *2nd generation* energy system models.**

Name	Type	Language	Licence	Purpose	Reference
Balmorel	M	GAMS	ISC Licence	ED,IN	[16]
Calliope	MG	Python	Apache-2.0	ED, IN	[18]
EMMA	M	GAMS	CC-BY-3.0	ED,IN	[19]
Dispa-SET	M	Python	EUPL-1.1	UC	[2]
OSeMOSYS	MG	MathProg	Apache-2.0	ED, IN	[17]
PyPSA	MG/F	Python	GNU GPL 3	ED,IN,PF	[20]
SciGRID	M	Python	Apache-2.0	PF	[21]
TEMOA	MG	Python	GNU GPL 2	IN	[22]

ED=Economic Dispatch, IN=Investment, PF=Power Flow, UC=Unit Commitment,
F=Framework, MG=Model Generator, M=Model

The approach of the framework presented in this paper differs from existing ones in various ways. Open source model generators like *calliope* presented by Pfenninger and Keirstead [18] and the toolbox *OseMOSYS* presented by Howells et al. [17] are designed to build specific models of one model family or type by the use of predefined sets of equations (e.g. bottom up linear optimization based models). Some of the projects listed in Table 1 include several model generators for different purposes that may be combined. For example the *PyPSA* project provides functionalities for non-linear power flow, linear optimal power flow and investment optimization [20]. With a standardized input-output interface across these functionalities *PyPSA* is located between a model generator and a framework. However, *PyPSA* was designed primarily for power system analysis with a focus on comprehensive power flow modelling. However, modelling multi-sectoral problems is also straightforward with *PyPSA*. Nevertheless, up to now *PyPSA* does not include any libraries for input data calculation such as wind or photovoltaic time series.

In addition to other projects, the proposed framework tries to encompass valuable model generator approaches while going one step beyond to facilitate

the modelling of complex energy systems. It provides a rich set of tools that can be understood as encapsulated sub-frameworks and allows to model highly integrated energy systems. At the current state, the toolset includes optimization libraries as well as tools to simulate feed-in from renewable energy sources or local heat demand for a specific region. Furthermore, the entire framework is embedded in a surrounding infrastructure for pre- and post-processing of data. This enables a complete modelling workflow for different problem types within one environment.

3. Architecture: The Framework

3.1. Genesis

The framework has been created as a community project to take advantage of similar modelling activities and avoid redundant work. Due to the different research backgrounds of the collaborators, oemof has not been designed for one particular application or research question. Therefore, it is neither focused on a specific analytic or mathematical approach nor restricted to a certain spatial or temporal resolution. Instead, the primary idea was to create a software that can be used to build a variety of energy system models by simplifying recurring processes in the field of energy system modelling. Hence, oemof was not designed for end-users but for energy system modellers with profound understanding of real world energy system problems and their possible abstracted mathematical representation.

One challenge in the development process was the consolidation of existing models from different institutions with a similar but partly different research focus into a common software project. To satisfy requirements of collaborating groups and projects, the framework needed to be flexible and adaptable. Furthermore, the produced code had to be well documented in order to be accessible and easily modifiable for contributors of various institutions. The general consensus of the project was to adhere to high scientific standards. Therefore, the project follows a strict open source and non-proprietary software development.

This collaboration of various researchers with different scopes and backgrounds led to the development of a framework with a common and generic basis (Section 3.2) consisting of layer structured set of tools and sub-frameworks (Section 3.3).

3.2. Underlying concept

The main feature of the framework is the separation of an energy system's topological description and its actual calculation based on a concrete parametrisation. Due to this property, oemof may even be used to incorporate other models and model generators with other programming languages in the future. To achieve this, a generic concept which constitutes the foundation of all oemof libraries has been developed. In this concept an energy system is represented as a network consisting of nodes and flows connecting these. Nodes N are subdivided into buses B and components C. When representing an energy system, an additional constraint that buses are solely connected to components and vice versa is imposed. Components are meant to represent actual producers, consumers or processes of the energy system while buses are meant to represent how these components are tied together. Flows are used to represent the inputs and outputs of a component.

An energy system that is represented in such a way can be mathematically described using concepts from graph theory by looking at it as a bipartite graph G. The mathematical formulation of this graph in its general form is given in Equation 1. A more detailed description of this concept with its theoretical foundation has been published by Wingenbach et al. [23]. The nodes of the graph are partitioned into buses B and components C while flows act as the directed edges E of the graph.

$$G := (N, E)$$
$$N := \{B, C\}$$
$$E \subseteq B \times C \cup C \times B$$
$$C^+ \subseteq C$$
$$C^- \subseteq C$$
$$T \subseteq C \tag{1}$$

Moreover the components can be subdivided into sources C^+, sinks C^- and transformers T:

1. *Transformers* have inflows and outflows. For example a gas turbine consumes gas from a gas bus and feeds electrical energy into an electricity bus. The relation between inflow and outflow can be specified in the form of parameters, for example by specifying the transfer function or an efficiency factor.
2. *Sinks* only have inflows but no outflows. Sinks can represent consumers of which households would be an example.
3. *Sources* have outflows but no inflows. For example, wind energy or photovoltaic plants but also commodities can be modelled as sources.

A similar, purely mathematical formulation of multi-commodity network flow optimization models for dynamic energy management has been illustrated by Zeng and Manfren [24]. Furthermore, related structures of energy systems can also be found in different energy models [17, 12, 25]. These publications demonstrate that using a graph is an intuitive way of representing an energy system. A graphical representation of how to describe an arbitrary energy system using this network structure is shown schematically in Figure 1.

Based on the described concept, oemof provides basic components which can be used directly while also facilitating the development of more specific components built upon the basic ones. Being an openly available, uniform, sector-independent framework that provides a consistent object-oriented imple-
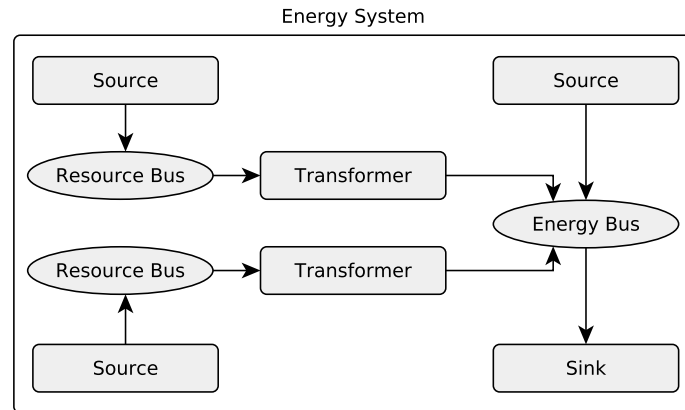
Figure 1: Schematic illustration of an energy system represented as an oemof network.

mentation of the generic network structure makes oemof a valuable contribution to the energy modelling and especially the energy framework family.

3.3. Implementation

The framework and its underlying concept are implemented using an object-oriented approach in the high-level programming language Python. It is published under the GNU GPL3 licence and consists of different libraries that can be grouped according to their mutual compatibility, resulting in the layer structure shown in Figure 2.

Four main layers can be used to categorize the libraries:

1. At the *core* layer the generic graph structure is implemented. At this layer the basic application programming interface (API) is defined. Additionally, the description of an energy system as well as the format of input and output data for core-objects is defined.

2. The *namespace* layer contains associated libraries that are based on the core APIs and inherit from *core*-objects. These could be libraries for cost or power-flow optimization or other purposes.
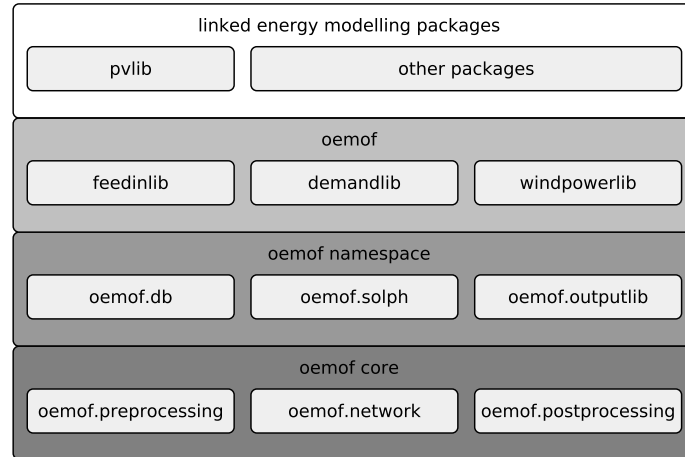
Figure 2: Libraries located on different oemof layers.

3. The *oemof* layer contains libraries associated with oemof. It is not based on a cross-level API. However, it is developed within the framework and consequently follows common development rules (Section 3.4). These libraries can either be used independently or combined with all other libraries.

4. Finally, the outermost *linked packages* layer includes open source projects known to be used by or to use libraries from the layers below. These open source projects complete the toolbox for energy modelling. With this layer the advantage of synergies with energy related open source projects that link to or are linked to the framework is used.

The network concept has been implemented at the *core* layer in form of a class hierarchy which is sketched in Figure 3. The root elements of this class hierarchy are *Node*, *Flow* and *EnergySystem*. *Node* is the abstract base class for *Bus* and *Component*, which are used to represent nodes in the bipartite graph representing the energy system. Furthermore, components are subdivided into *Source*, *Sink* and *Transformer* classes depending on how they are connected to *Bus* objects. Objects of the class *Flow* represent the directed edge between two

nodes, i.e. the connection between a *Bus* and a *Component* object. Moreover, the class *EnergySystem* serves as a container for nodes and may hold additional information about the energy system.
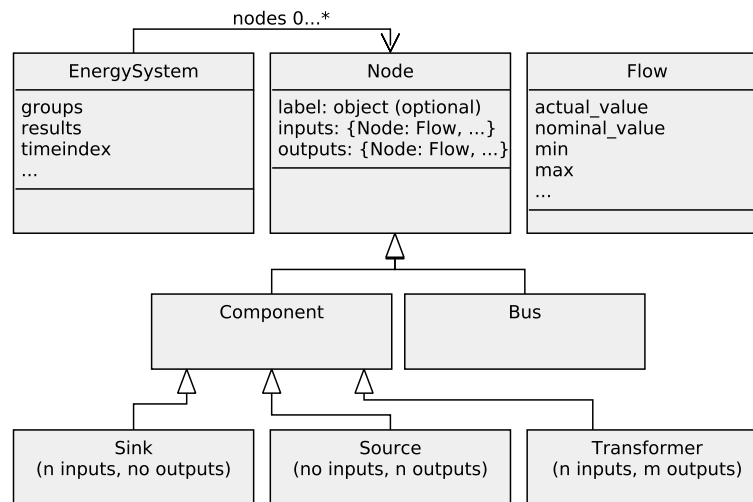
Figure 3: UML diagram of oemof core classes.

All basic energy system components such as energy demands, (renewable) energy sources and transformers between different energy buses can be modelled by means of these basic classes. Additional components that introduce new features can be added via inheritance. If sub-classing is not suitable, new classes can be created and used together with the core classes. As an example, the *solph* library introduces a storage class with different individual parameters.

The initially outlined main framework feature of separating the description of the energy system from its simulation is reflected by the introduction of the *Flow* class which is separate from the *Node* class hierarchy. Objects of this class hold information about the flow between two nodes such as maximum available transfer capacities of power line flows or whether the amount of a certain flow is fixed and if so, its value. As an example for the provided generic flexibility, objects of this class are used in the *solph* library to build inter-

temporal constraints for different kinds of energy system optimization problems such as combined heat and power modelling or unit commitment.

A representation closer to real-world energy systems can be created by the *EnergySystem* class. The class serves as a container for the above mentioned elements and provides the possibility to add extra information such as grouping structures or optimization parameters. Additionally, it provides interfaces to save and restore entire energy systems and to process results. This allows for an intuitive handling of energy systems by treating them as an own entity.

Overall, an implementation within the high level programming language Python has the advantage of a rich set of external libraries usable for scientific computing. Oemof itself makes heavy use of external modules for optimization problem [26] and data handling [27]. Additional packages for data pre- and post-processing or visualization can be included easily. The structure empowers the user to decide the extend to which internal or external packages should be used in his applications.

3.4. Documentation, collaboration and testing

'A critical part of any piece of software is the documentation' has already been stated by Greenhall and Christie [28]. This is of particular importance for open source projects with many users and a changing developer base. With the objective of a profound documentation in all stages and formulation of general nomenclature, a documentation approach on four different levels similar to Howells et al. [17] is followed:

1. *Comments inside the code* are used to explain non-intuitive lines of code to new developers and interested users at the lowest level.
2. *Docstrings* located inside the source code describe the API, i.e. how to use the various classes, methods, and functions.
3. *Higher level descriptions* provide the user with additional information about the possible interactions between different libraries or application-specific usage information. These manuals are located inside the repository and are therefore shipped with the source code.

4. *Examples* provide an additional source of documentation that is particularly useful to new users.

Keeping such detailed documentation consistent and up to date across continous releases comes at the expense of a high maintenance effort. Nevertheless, it enables a high degree of transparency which is of special importance if external users want to understand the internal logic of a model in order to reproduce scientific results.

As oemof is an open source community project, a common platform for collaboration is needed. Similar to Greenhall and Christie [28] as well as other open source energy modelling projects, oemof uses GitHub for collaboration, code hosting and bug-tracking which allows for an easy copying and forking of the project. GitHub is based on the version control system *git* and code can be developed in parallel on different branches. In order to ensure an effective branching strategy and release management, a well established git workflow model [29] is set as standard for all developers. Contributions to the code base are managed in form of pull requests which allow for an openly discussed review process of potential changes. Moreover, code changes are checked for conflicts before being merged back into the development branch by the respective developer in charge of the affected library.

In order to test oemof's functionality in case of changes to multiple parts of the code base, *unit tests* are applied. During the testing process, all integrated application examples are run and the created results are checked against stored historical results. Only if all examples run without errors a pull request is merged back into the development branch. This procedure ensures the functionality even if major changes to the code base are applied from one release version to another.

4. Usage: Applications

The framework is not designed to constitute a standalone executable. Instead, the oemof libraries are meant to be used in combination with input data,

parameters and possibly other Python libraries to build *applications*. Applications may depict a specific energy system or provide an abstraction layer to a customisable modelling software (sub-framework).

4.1. Application development

Generally, applications can be developed highly individually by the use of one or more framework libraries depending on the scope and purpose. Figure 4 illustrates the typical process of building an application. In this context, the libraries are grouped according to their domain when viewed from a potential user's perspective and not according to their layer in the cosmos like in Figure 2.
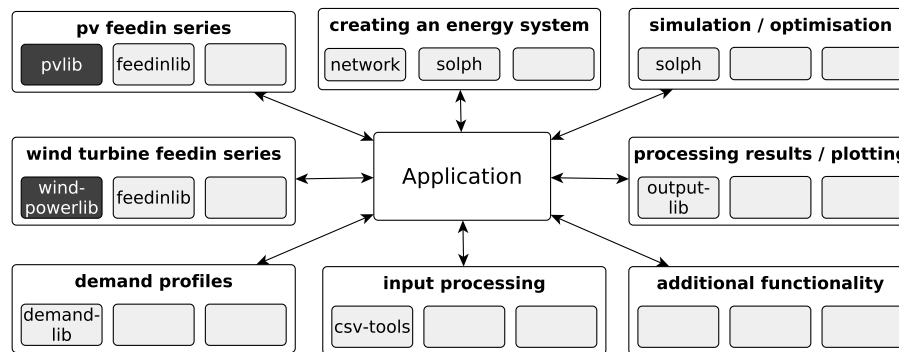
Figure 4: Building an application based on libraries of the oemof cosmos and external libraries (dark grey).

When developing applications, specific components can be substituted easily according to the functionality needed. This allows for individual application development and provides a high degree of freedom for the developer. Such flexibility is particularly relevant in scientific working environments with spatially distributed contributors and constantly required adjustments for upcoming research questions. The modelling processes can take various forms ranging from a few plain steps in a standalone Python file to complex procedures bundled in a new Python library based on oemof.

However, regarding the modelling workflow, all applications have some major steps in common and throughout include all required data pre- and post-

processing. First of all, an empty energy system object is created. This object acts as a container for the nodes and carries information such as the time resolution. The energy system object may also hold different scenarios representing different developments of the system. Additionally, methods to handle nodes are provided. The next step is the instantiation of nodes and flows of the modelled energy system which are added to the existing energy system instance (population of energy system). Subsequently, the results of the energy system can be computed by simulating or optimizing the system. For the simulation/optimization, objects created from *namespace*-layer classes as well as *core*-layer objects may be used. However, objects of different *namespace* libraries might not be compatible with each other. Nevertheless, all objects will be compatible with the functionalities of the *core*-layer. Finally, results can be processed with the output library of oemof. The *oemof-outputlib* makes it easy to get different views of the results and plots based on a uniform output data format.

Despite their specific differences, oemof applications share a similar structure and terminology. Therefore, it is easier for developers to comprehend oemof based models than completely different, arbitrary other models.

4.2. Existing applications

The framework has already been used to build comprehensive applications for different research projects (s. [30, 31, 32, 33, 34, 35]). In addition, oemof is also used actively in teaching by some institutions in order to gain insights into complex energy systems. An example for such a system modelled as an oemof application is illustrated in Figure 5. In the following, selected oemof applications are described to illustrate the broad range of applications.

The renewable energy pathways simulation system (*renpassG!S*) [36] is a bottom-up fundamental Western European electricity market model. Future scenarios of the power plant dispatch and price formation in Germany and its interconnected neighbouring countries can be modelled based on operational and marginal costs and the assumption if an inelastic electricity demand. Currently,
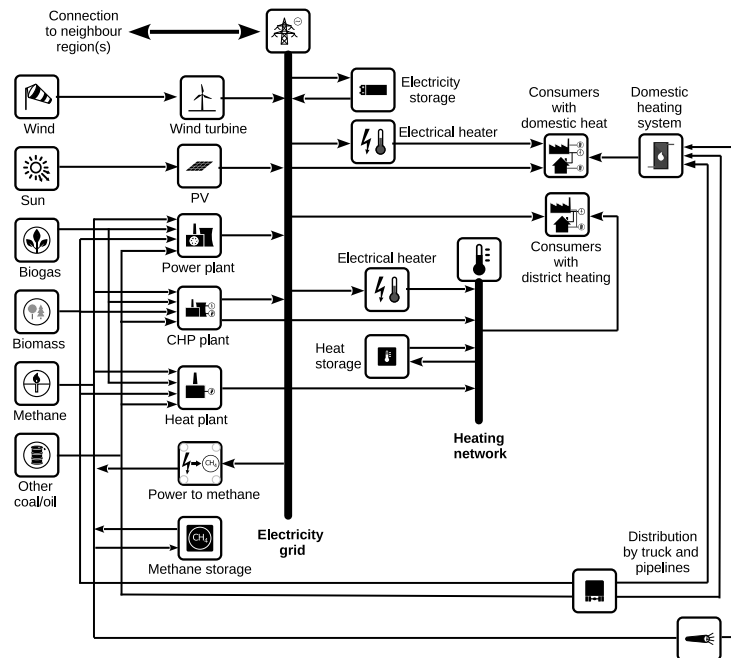
Figure 5: Representation of a complex energy system within an oemof application.

a similar spin-off model is adapted to the requirements of the Middle East and North Africa (MENA) region.

The *openMod.sh* model is flexible software tool that is strongly based on oemof's underlying concept [33]. This model has been applied in participative workshops for the development of regional climate protection scenarios. The combination of a graphical browser-based user interface combined with an open source modelling approach enhances the modeller-decision maker interface. As the underlying concept has been extended to a generic database concept this example shows that oemof's concept can not only be used for the simulation of systems but also for its representation.

An example for the flexible extension of oemof on the application level is demonstrated within the Heating System Optimization Tool (*HESYSOPT*) [37]. In this application, detailed heating system components are modelled with mixed integer linear programming techniques that are based on basic oemof

functionalities.

As a third example, *reegis*^{hp} [38] models heat and power systems on a local scale. The objective is to evaluate district heating and combined heat and power technologies in energy systems with a high share of renewable resources from an environmental and economical perspective. The local system is connected in terms of electricity to a national model based on the idea of the model *renpassG!S* [36] which is extended to include the heating sector.

These applications illustrate the flexibility of oemof and the extent of the potential user group not only content- and resolution-wise, but also concerning the level of involvement. It is possible to build a full-scale energy system model adapted to the user's needs by just employing existing functionalities. Moreover, different models can be combined and adapted at low effort to create tools for specific purposes. Overall, this enables users to answer challenging research questions within a single framework.

5. Conclusion

The paper presents the Open Energy Modelling Framework (oemof) as a contribution to the scientific modelling community. With its philosophy and concept it represents a novel approach in energy system modelling.

While other existing frameworks and model generators are designed to construct specific models of one model family or type, oemof can rather be understood as a structured set of tools and sub-frameworks to construct comprehensive energy system models. Therefore, libraries for computing time series of renewable energy sources feed-in or electricity demand profiles as well as optimization libraries for dispatch and investment models are included in the framework. Different levels of mutual compatibility among libraries are handled in a layer structure. Due to this structure, oemof differs from a simple toolbox as it relates the parts inside the framework around possible applications in energy system modelling.

One central feature of oemof is the generic graph based foundation which

has been implemented using an object-oriented approach in the high level programming language Python. The cross-institutional collaborative development of the framework has triggered a process towards this common and generic structure. This concept highlights to distinguish between the description of an energy system with its components and its subsequent computation based on a specific mathematical approach. Moreover it lays the foundation for universal representation of multi-sectoral energy systems at different scales. Another important feature of oemof is its strict open source and non-proprietary philosophy. This philosophy, the underlying concept and the extensive documentation allows new developers to adapt or extend the framework easily and leverage features of other scientific Python libraries. With these properties, the project is geared towards new developers and users and thereby a continuous future development process.

The framework has already been successfully applied within different research projects across several institutions. Among others, existing oemof applications include electricity market models, detailed technical unit commitment models for district heating systems and sector coupled regional energy system models. Essentially, energy systems ranging from distributed or urban ones up to a national scale may be modelled, making the framework a multi-purpose modelling environment for strategic energy analysis and planning.

Acknowledgements

The group of authors of this paper is not identical with the oemof developer group. Since oemof is a collaborative project, all developers can be found on the contributors page of each repository on GitHub. The project was initially created by the following institutions:

- Europa-Universität Flensburg (EUF)

- Flensburg University of Applied Sciences (HFL)

- Otto von Guericke University Magdeburg (OVGU)

- Reiner Lemoine Institut Berlin (RLI)

Special thanks and acknowledgement is owed to all people who have been supporting the project since its starting phase. In particular we want to thank Birgit Schachler (RLI), Caroline Möller(RLI), Martin Söthe (ZNES), Wolf-Dieter Bunke (ZNES) and Steffen Peleikis for their individual contributions throughout different stages.

References

[1] S. Pfenninger, A. Hawkes, J. Keirstead, Energy systems modeling for twenty-first century energy challenges, Renewable and Sustainable Energy Reviews 33 (2014) 74–86.

[2] S. Quoilin, H. Gonzalez, Zucker, Modelling Future EU Power Systems Under High Shares of Renewables: The Dispa-SET 2.1 open-source model, JRC TECHNICAL REPORTS, 2017. doi:`dx.doi.org/10.2760/25400`.

[3] D. R. Biggar, M. Reza Hesamzadeh, The Economics of Electricity Markets, John Wiley & Sons, 2014.

[4] C. Harris, Electricity Markets: Pricing, Structures and Economics, John Wiley & Sons. Second Edition, 2006.

[5] A. J. Wood, B. F. Wollenberg, G. Sheble, Power Generation, Operation and Control, John Wiley & Sons. Third Edition, 2013.

[6] D. Kirschen, G. Strbac, Fundamentals of Power System Economics, John Wiley & Sons, 2004.

[7] S. Pfenninger, J. DeCarolis, L. Hirth, S. Quoilin, I. Staffell, The importance of open data and software: Is energy research lagging behind?, Energy Policy 101 (2017) 211 – 215.

[8] S. Pfenninger, Energy scientists must show their workings, Nature 542 (2017) 393.

[9] Openmod-Initiative, openmod - open energy modelling initiative, http://www.openmod-initiative.org/, 2017. (accessed 26.04.2017).

[10] European Commission, Horizon 2020 - the eu framework programme for research and innovation, https://ec.europa.eu/programmes/horizon2020/, 2017. (accessed 27.04.2017).

[11] L. M. Hall, A. R. Buckley, A review of energy systems models in the UK: Prevalent usage and categorisation, Applied Energy 169 (2016) 607–628.

[12] R. Loulou, G. Goldstein, K. Noble, Documentation for the MARKAL Family of Models, Technical Report, Energy Technology Systems Analysis Programme, 2004.

[13] L. Schrattenholzer, The Energy Supply Model MESSAGE, IIASA Research Report, IIASA, Laxenburg, Austria, 1981. URL: `http://pure.iiasa.ac.at/1542/`.

[14] European Commission, POLES - Global energy model, https://ec.europa.eu/jrc/en/poles/model, 2017. (accessed 27.04.2017).

[15] Aalborg University, Energy Plan - Advanced energy system analysis computer model (Version 12.5), http://www.energyplan.eu/, 2016. (accessed 05.05.2017).

[16] H. F. Ravn, The Balmorel Model: Theoretical Backround, http://balmorel.com/images/downloads/The-Balmorel-Model-Theoretical-Background.pdf, 2001. (accessed 04.05.2017).

[17] M. Howells, H. Rogner, N. Strachan, C. Heaps, H. Huntington, S. Kypreos, A. Hughes, S. Silveira, J. DeCarolis, M. Bazillian, A. Roehrl, OSeMOSYS: The Open Source Energy Modeling System: An introduction to its ethos, structure and development, Energy Policy 39 (2011) 5850–5870.

[18] S. Pfenninger, J. Keirstead, Renewables, nuclear, or fossil fuels? Scenarios for Great Britain's power system considering costs, emissions and energy security , Applied Energy 152 (2015) 83–93.

[19] L. Hirth, The european electricity market model emma model documentation, http://neon-energie.de/EMMA, 2016. (accessed 27.04.2017).

[20] T. Brown, J. Hörsch, D. Schlachtberger, PyPSA: Python for power system analysis version 0.8, http://pypsa.org, 2017. (accessed 26.04.2017).

[21] C. Matke, W. Medjroubi, D. Kleinhans, SciGRID - An Open Source Reference Model for the European Transmission Network (v0.2), 2016. URL: `http://www.scigrid.de`.

[22] K. Hunter, S. Sreepathi, J. F. DeCarolis, Modeling for insight using Tools for Energy Model Optimization and Analysis (Temoa), Energy Economics 40 (2013) 339–349.

[23] C. Wingenbach, S. Hilpert, S. Günther, The core concept of the Open Energy Modelling Framework (oemof), in: Environmental Informatics – Current trends and future perspectives based on 30 years of history, 2016.

[24] D. Zeng, M. Manfren, Multi-commodity network flow models for dynamic energy management – Mathematical formulation, Energy Procedia 14 (2012) 1380–1385.

[25] J. Richter, DIMENSION - A Dispatch and Investment Model for European Electricity Markets, 2011. EWI Working Papers 2011-3.

[26] J.-P. W. Willian Hart, Carl Laird, D. L. Woodruff, Pyomo - Optimization Modeling in Python, Springer, 2012.

[27] W. McKinney, Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython, O'Reilly, 2012.

[28] A. Greenhall, R. Christie, Minpower: A power systems optimization toolkit, in: 2012 IEEE Power and Energy Society General Meeting, 2012, pp. 1–6. doi:`10.1109/PESGM.2012.6344667`.

[29] VanDriessen, A successful git branching model, http://nvie.com/posts/a-successful-git-branching-model, 2010. (accessed 31.10.2016).

[30] C. Möller, K. Kuhnke, M. Reckzugel, H.-J. Pfisterer, S. Rosenberger, Energy storage potential in the Northern German region Osnabrück-Steinfurt, IEEE, 2016, pp. 1–7. doi:`10.1109/IESC.2016.7569497`.

[31] U. P. Müller, I. Cussmann, C. Wingenbach, J. Wendiggensen, AC Power Flow Simulations within an Open Data Model of a High Voltage Grid, Springer International Publishing, Cham, 2017, pp. 181–193. doi:`10.1007/978-3-319-44711-7_15`.

[32] M. Degel, M. Christ, J. Grünert, L. Becker, C. Wingenbach, M. Soethe, W.-D. Bunke, K. Mester, F. Wiese, VerNetzen: Sozial-ökologische und technisch-ökonomische Modellierung von Entwicklungspfaden der Energiewende. Projektabschlussbericht, IZT Berlin, Europa-Uniersität Flensburg, Deutsche Umwelthilfe e.V., 2016.

[33] C. Wingenbach, S. Hilpert, S. Günther, openMod.SH - Ein regionales Strom-Wärme-Modell für Schleswig-Holstein basierend auf open source und open data, in: 14th Symposium on Energyinnovation, Technical University Graz, 2016.

[34] C. Kaldemeyer, C. Boysen, I. Tuschy, Compressed air energy storage in the german energy system – status quo and perspectives, Energy Procedia 99 (2016) 298 – 313. 10th International Renewable Energy Storage Conference, IRES 2016, 15-17 March 2016, Düsseldorf, Germany.

[35] O. Arnhold, M. Fleck, K. Goldammer, F. Grüger, O. Hoch, B. Schachler, Transformation of the German energy and transport sector - a national analysis, in: 2. Internationale ATZ-Fachtagung: Netzintegration der Elektromobilität 2017, Berlin, 2017. (Paper accepted).

[36] ZNES, renpass-gis (Renewable ENergy PAthway Simulation System capable of working with GIS data), https://github.com/znes/renpass_gis, 2016. (accessed 31.10.2016).

[37] S. Hilpert, HESYSOPT - An open source tool to support district heating system flexibilisation., in: V. Wohlgemuth, F. Fuchs-Kittowski, J. Wittmann (Eds.), Environmental Informatics – Current trends and future perspectives based on 30 years of history, Shaker, 2016, pp. 361–366.

[38] U. Krien, reegis - An oemof application to model local heat and power systems, https://github.com/rl-institut/reegis_hp, 2016. (accessed 31.10.2017).