Towards a Classification Scheme for Co-Simulation Approaches in Energy Systems

Florian Schloegl, Sebastian Rohjans,
Sebastian Lehnhoff, Jorge Velasquez, Cornelius Steinbrink
OFFIS - Institute for Information Technology
Oldenburg, Germany
Email: [surname]@offis.de

Peter Palensky
Technical University of Delft
Delft, the Netherlands
Email: p.palensky@tudelft.nl

***Abstract*—Simulations become more and more crucial in the field of future energy systems. This is caused by the increasing complexity of energy systems that consist of a variety of subsystems such as supply infrastructures, production, consumption, markets, communication, meteorology etc. Co-simulation tools provide the possibility to combine models of these subsystems and run them in a coordinated simulation. However, such simulations become more and more complex, making it improbable that the user of a simulation is the same person that develops the simulation system. To facilitate the communication between users and developers of co-simulation tools and to help the user to find the suitable software for his purpose, the authors suggest a typification of co-simulation tools. This is done by identifying the most relevant attributes each specified by a set of possible configurations. The utilization of the developed schemed is demonstrated by applying it to the mosaik co-simulation framework.**

I. Introduction

Smart Grids[1] offer opportunities but also pose challenges to future power and energy systems. The penetration of the electrical grid with ICT (Information and Communication Technologies) allows a much more flexible operation thus facilitating the integration of renewable energy sources, but also creates opportunities for new business models [1]. On the other hand the already complex power supply system becomes even more complex, making the development of new components and algorithms difficult. The fact that the power supply system is a critical infrastructure forbids trial-and-error-approaches in the running system. A well-established approach for the analysis of Smart Grids and the development of Smart-Grid-components is the use of simulations [2] [3]. These simulations may be purely software-based or span from software to hardware simulations by integrating algorithms or controllers on realistic platforms "in-the-loop" [4].

Simulation of Smart Grids requires the cooperation of specialists from many different fields: electrical engineers, computer scientists, economists, environmental physicists etc. Experts in these disciplines develop models for different parts of the future Smart Grid, which have to be integrated into a simulation environment. There are many models to choose from, e.g., highly sophisticated simulation models for electrical grids, or communications system models [5]. When simulating these domain-specific models potential techniques may be distinguished into four categories (see Fig. 1):

- Simulating a given model in a given model of a specific representation (e.g. ordinary/partial differential equations, discrete automata, time series) within a dedicated runtime environment/solver is most common.
- To speed up computational time, models may be divided and run in parallel on multiple solvers. Although multiple models are executed in this *parallel simulation* there is only one type model representation.
- In *hybrid simulation* models of different representation are integrated into a single runtime environment.
- In *co-simulation* models of different representation are executed/solved in individual runtime environments. Synchronizing and orchestrating this potentially complex setup of heterogeneous models and their individual solvers is a particular challenge in co-simulation.

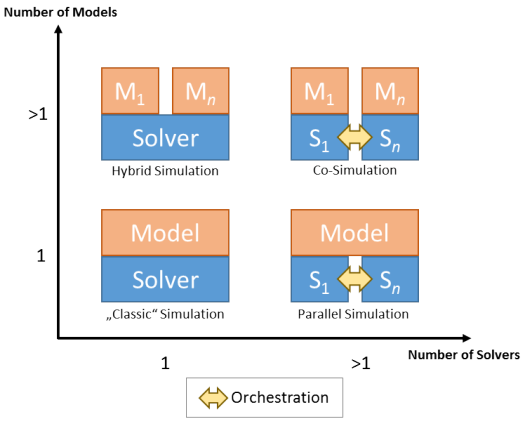

Fig. 1. Simulation categories

Co-simulation allows to reuse models and enables experts from one domain and discipline to concentrate on their respective field of expertise. The modeling can be done on the subsystem level without having the coupled problem in mind. The coupled simulation is carried out by running the

[1]Following the IES's definition of a Smart Grid: electric power system that utilizes information exchange and control technologies, distributed computing and associated sensors and actuators, for purposes such as:

- to integrate the behaviour and actions of the network users and other stakeholders,
- to efficiently deliver sustainable, economic and secure electricity supplies

http://www.electropedia.org

978-1-4799-7736-9/15/$31.00 ©2015 IEEE

subsystems in a black-box manner. During the simulation the subsystems will coordinate implicitly through exchange of data [6]. There is an obvious trade-off between the independent development of subsystem models followed by their comfortable integration in co-simulation and performance/efficiency in simulating complex systems (of systems) in parallel or hybrid simulation.

A. Motivation

As the co-simulation environments evolve and become more powerful and complex, the people developing simulation tools are no longer the same as those that are applying them. The dialog between developers and users is not always simple. A common misunderstanding by the users is that a part of the system once modeled can be applied to all kinds of simulation-purposes [7]. A model is a simplified representation of a real world object or system. It reproduces the relevant aspects of that object or system for systematic analysis that are driven by specific use cases. What these relevant aspects are, depends on the kind of analysis that is to be executed [7]. A model of a biogas plant developed to analyze the gas production depending on the substratum will not be used to simulate the overall revenues from this biogas plant simply because the effort for the computation would be too high. Thus, for an effective development and use of co-simulation tools, it is necessary to analyze the application of these tools.

The common approach to analyze the application of software is to identify use cases and derive requirements from them [8]. This approach is inappropriate here as one has to deal with a huge number of use cases and, more importantly, as Smart Grid simulation is a new field and subject of ongoing research, not all use cases are known yet [9] [10]. For this reason a more general approach is required, i.e. a classification of use cases, which allows a typification of Smart Grid co-simulation and Smart Grid co-simulation tools.

B. Goals

Typical questions in the dialog between users and developers of co-simulation tools are [11]: What is the right tool for my research question? or: In what direction should I develop my tool? The goal of this paper is to facilitate this discussion by providing some basic definitions and a classification scheme for simulations and simulation tools. This classification scheme is then applied to the co-simulation framework mosaik[2] as an example.

II. Approach

The software engineering standard approach when developing software tools is to start with analyzing use cases and obtaining the requirements of the software product [8]. As explained in the introduction, this approach is difficult in this particular case, mainly because the use cases are subject of ongoing research. That's why a more generic approach has been chosen: First an understanding of actors in the Smart Grid context has to be gained thus getting an impression of the questions that have to been answered by using simulations. With this basic understanding suitable categories to type co-simulation tools can then be found.

[2]http://mosaik.offis.de/

A. Analysis of Users

To analyze use cases for simulations in Smart Grids, it is important to have a basic understanding of motives and interests of the actors in energy and power systems. A straightforward approach to gain such an understanding is to answer the following three questions:

- Which are the most important actor groups in Smart Grids?
- What are the questions these actors have?
- How can these questions be answered by using simulations?

1) Grid Operator: How does the roll-out of new components like distributed energy sources (DER), storage units, tap transformers etc. affect the grid operation? Do I get problems with certain operation modes, e.g., the simultaneous reaction of production units to market signals [12]? What are the measures (use of flexibilities, grid expansion) I can take to optimize grid operation? What are the possibilities to optimize grid operation created by ICT-components?

Focus: simulation of different scenarios for the grid expansion with DER and analysis of their impact; development and evaluation of novel concepts for grid operation (agent-based control concepts, self-healing, self-organization, self-configuration) [13].

2) Producer of ICT-Components and -Systems: What are the requirements resulting from grid operation or VPP-operations concerning communication (bandwidth, topology, latency, reliability, security), monitoring and control, scheduling, accounting, balancing group management, transparency?

Focus: requirement analysis and optimization of IT-architecture based on simulation of different use cases [14].

3) Producer of Grid-Components and DER: What do I have to do to get my products "Smart Grid ready"?

Focus: simulation of use cases to analyze the requirements for control systems, communication (interfaces) and the operation of grid components [14]; test and evaluation of hard- and software components (Hardware-in-the-Loop, HiL) [4]; certification of hard- and software components.

4) Virtual Power Plant Operator, Energy Supplier: What are new possibilities for business cases created by Smart Grids? Is my business case viable? How can I optimize my revenues?

Focus: development and validation of business cases in simulations [15].

5) Policy and Regulation: How do I have to design legal and economical frameworks (Grid codes, market design, regulation etc.) to reach politico-economical and/or technical goals?

Focus: evaluation (cost-benefit-analysis, grid stability, quality of supply, etc.) of the impacts of changes of rules and regulations [9]; evaluation of the impacts of the introduction of new technologies [16].

6) Education and Science: What are the elements of the Smart Grid? How do these elements interact? How do I design a Smart Grid?

Focus: studying of and experimenting with simple and easy-to-use demo-simulators; expansion of existing and development of own scenarios [17] [18].

B. Categories

The following subsections present the six characteristics that we consider (based on a comprehensive literature research considering for instance [19], [20], [21], [5], [22], and [23]) the most important to describe co-simulation frameworks.

1) Time Resolution: The phenomena in electric power grids are of different physical origin and fall therefore into different categories of time constants. Traditionally three fundamental categories [20] are named (please note that the borders are fuzzy):

- Steady-State Range (seconds and above): the system is in equilibrium and phenomena like optimal power flow are studied.
- Electro-Mechanic Range (sub-seconds): the (mechanical) generators are dynamically interacting, and frequency transients, stability and short circuit behavior is analyzed.
- Electro-Magnetic Range (milliseconds and below): electromagnetic waves are the dominating mechanism; fast transients, dynamic voltage problems and stability are examples.

The components of the power system and their behavior typically also fall into one of these categories. FACTS (Flexible-AC-Transmission-System) devices for instance are fast switching elements that create electro-magnetic waves. An under-load-tap-changing transformer, on the other hand, is a slow component working above seconds.

2) Synchronization: The concept of co-simulation used in this paper is limited to simulators running in parallel. When running two or more simulators in parallel, there will always be the need to synchronize them. Providing functionalities to ensure a coordinated run of simulators is a central feature of co-simulation tools. The time resolution of the simulators to be coupled may vary [21]:

- Continuous[3]: the simulator is able to produce output for every point in time. This is the case if the underlying modeling is done by using differential equations or if hardware simulators like OPAL-RT[4]-simulators are used.
- Fixed Step: the simulator produces output for discrete points in time with a constant step size. This is commonly the case if physical systems are modeled that are too complex for an analytical solution so that numerical algorithms have to be used.
- Variable Step: the simulator produces output for discrete points in time, the interval between two steps may vary. The difference to an event driven simulation (s. below) is that the step length have to be known at least one simulation step in advance.
- Event Driven: the simulator produces output at discrete points in time with random intervals between the steps. This is the adequate modeling for systems that contain discrete dynamics like the switching of circuit breakers or the sending or receiving of data packets [5].

There are different strategies to combine these types of simulators [21]. Continuous hardware simulators can be coupled physically. Otherwise continuous simulators can be treated as fixed step simulators with very small step size.

When combining fixed step simulators, these may have different step sizes. In this case it is important, that the employed co-simulation tool supports this [23].

Finally it may become necessary to combine event driven simulators with fixed step simulators. In this case inaccuracies may arise when events of the event driven simulator do not fall exactly into synchronization points with the fixed step simulator. The simple approach is to reduce step size of the fixed step simulators to reduce these inaccuracies. A balance between synchronization accuracy and performance has to be found [5]. If the fixed step simulator supports roll-back, it is not necessary to synchronize the simulators exactly. It is then possible to set the fixed step simulator back to the state when the event of the event driven simulator occurred [22].

3) Time Ratio: Time ratio describes the relation between simulation time and wall clock time:

- One important feature of simulations is that it is possible to run a scenario faster than wall clock time thus covering a larger time interval than it would be possible in, e.g., lab experiments. So in most use cases the aim is to run the simulation as fast as possible.
- However, when the simulation is combined with real components (e.g., HIL) the co-simulation tool has to be able to run the simulated part in wall clock time.
- In some cases where phenomena have to be studied very detailed (e.g., with a very high time resolution) or computing resources are limited, it becomes necessary to run the simulation slower than wall clock time.

A more elaborated overview over this aspect can be found in [5].

4) Simulation Type: In contrast to purely analytic methods, dynamic simulation implements the behavior of the system under research via some formal language, interpreted by a solver. Depending on the nature of the system, different types of description and solvers are in use. Before the digital computer, analogue simulation with electric circuits, representing the system, were the common method to analyze complex behavior and to do experiments where analytic methods were not appropriate. Nowadays all simulations are based on digitally executed models. Aside from the trivial case, one model with

[3]In this context continuous must not be understood in an "analogue way". Every model that is executed on computer hardware will be discretized. The processor frequency sets the limit for the step size.

[4]http://www.opal-rt.com/

a solver, they can be distinguished into the following, non-exclusive categories [24], [25]:

- Hybrid Simulation: the model is spread over two or more submodels. One solver works with all of them. The motivation for having the system described in different submodels might stem from specialized description languages for the individual aspects of the system.
- Co-Simulation: Similar to hybrid simulation, but several solvers interact, synchronize, and exchange variable values [26].
- Hardware-Supported Simulation: dedicated computer hardware is used to execute the model. This is typically motivated by performance reasons.
- Hardware-in-the-Loop (HIL) Simulation: real elements of the system are connected to a RT simulation and interact bidirectionally with it. These real elements can be physical hardware, software programs (e.g. implementation of control algorithms), and even humans.

5) Elements and Modeling: Modeling Smart Grids leads to the ability of describing the following four categories of submodels [19]:

- Continuous Processes: this includes all physical infrastructure like power lines, generators, heat pipes, machines, etc. Traditionally, these elements are described using differential equations.
- Discrete Processes and Events: all information and communication technology is based on discrete processes. Especially asynchronous events are of interest, since they might originate from the continuous domain (e.g., a threshold is reached, an alarm is generated) and also might feed back into the dynamics of the continuous parts (e.g., a switch changes the topology of the grid).
- Roles: the behavior of individuals, e.g., customers or market traders, might be considered in multi-agent sub-models or in game theoretic models.
- Statistical Elements: some submodels are of statistic or stochastic nature. Examples are observed reliability of components or weather phenomena.

Depending on the questions asked, some parts of the system can be simplified or even excepted from the dynamics by replacing them with static behavior or lookup tables.

6) Accessibility of Internal Model Structures: Depending on the knowledge of the system under investigation, the model can follow the systems internal structure more or less [27].

- Black Box: the case where only the behavior of the system can be observed and imitated is called a black box case. If testing, i.e. applying test inputs, is easy, model identification techniques might be used. A prerequisite of that is the choice of the model category (e.g., second order linear system). An alternative to assuming a certain model category/order is to use a universal model method and let the system learn (e.g., statistical models, neural networks, etc.). An example would be the electric load of a distribution grid feeder without knowledge of the individual loads.
- White Box: if the elements of the system are completely known and modeled in all its physical and functional parts according to the current art of science, we receive a white box model. An example is an electric circuit, whose elements and components are known. This also assumes that the physical and functional laws that govern the behavior of the connected elements are fully known [28].
- Gray Box: if parts of the system are known or if there is good reason to believe that the system consists of certain parts we speak of a gray box model. Some parameters might be known but some are not and need to be chosen or identified. Similar to the black box model, the choice of the parameters requires identification techniques. An example would be an electric compression chiller whose principles are clear, whose data sheet gives its elements and some parameters, but the detailed physical properties are not known.

It is clear that white box models are to be preferred, if the model is simple enough to be constructed and worked with. However, such models can easily get complex. Digital controls and software are state machine with immense complexity. It is impossible to identify its structure and behavior in a black box fashion. In most cases, only white box models seem to be appropriate for this important smart grid element.

In ever larger, complex and interdisciplinary setups for co-simulation white box models can easily become black box models since users that are proficient in one field may lack the expertise to analyze and competently modify or integrate a white box model from another field, e.g. an expert in power system modeling may not be able to understand a model of a social system in a white box manner.

7) Usability: As co-simulation tools evolve and become more and more complex, it gets less practical that the user of a simulation tool is also its developer. Domain experts wish to focus on their research questions and do not want to deal with the programming of their tools. For that, it is necessary to describe for which user group the tool is suitable. The user groups have to be distinguished not only by their field of expertise (power grid, communication, monitoring and control, economics, etc.) but also by their level of expertise (students, engineer, researcher, etc.). The usability (in terms of interfacing) of co-simulation tools can be divided into [29] [30]:

- Application Programming Interfaces (API) or
- Graphical User Interfaces (GUI).

As such features have to be developed for quite specific use cases, this leads in general to a decrease of flexibility. A trade off between these two qualities has to be found.

III. The mosaik Framework

A. Description

Mosaik is a co-simulation framework that enables the interaction between different simulation tools in order to

create a consolidated Smart Grid scenario [31], [32], [33]. One important feature of mosaik is that it is based on discrete-event simulation. This means that the execution of the implemented simulators is performed in an event-based manner. In addition, mosaiks main advantage is that it is capable of managing and integrating specialized (as well as multidisciplinary) simulation platforms allowing the creation of custom made scenarios according to the users needs (e.g., a renewable energy production scenario that incorporates market constrains and weather forecasting models). Furthermore, in mosaik simulators exchange data according to their required attributes, e.g., a grid and a household simulator share the common attribute power and are connected to each other, on this basis: the household can provide the power consumption information to the grid simulator. To achieve this, mosaik makes use of the following four basic components.

The mosaik *SimAPI* designates the communication protocol between the different simulators and mosaik. This interface is further divided into a low and a high-level API. Both interfaces utilize JSON encoded messages and network sockets. For the low-level API, the users themselves have to set up the socket connections and the JSON serialization required to connect the defined simulator to mosaik. In addition, the low level API is implemented independent of the utilized programming language (commonly referred to as language agnostic). In contrast, for the high-level API, the network related parts are already enclosed within mosaik. This facilitates the implementation of a specific simulator because it spares the user from defining all the network components needed for establishing the coupling between the simulators and mosaik. This high-level API is currently available for Python and Java. The SimAPI is implemented via the following functions: the *init()* call is used to start the simulators, the *create()* call is used to instantiate the different models implemented in the simulators, the *step()* call is a function that advances the simulator in time by updating the simulation state and returning the new time for the next step, and the *get_data()* call allows mosaik to retrieve the values of the attributes defined in *init()*.

The *scenario API* of mosaik allows the user to create simulation scenarios. In order to do this, it is necessary to define an environment that contains all the information related to the simulators that are going to be integrated in mosaik. This is called the creation of a World. This environment holds information about the time frame of the simulators. Once the World is created, it is possible to start the simulators with the function *World.start()*. The next step is to instantiate the models associated with the simulators, in order to create a set of entities. Once the entities are created from a model, they are connected to each other via a defined rule-based topology. This means, that the user is able to set a series of connection criteria and mosaik performs accordingly. After this, it is possible to run the defined scenario, which leads to execution of the simulation.

The *Simulator Manager* of mosaik is the component responsible for communicating and starting the processes of the implemented simulators. As stated before, the communication between mosaik and the simulators is performed via JSON encoded messages and network sockets. In addition, there are three different management possibilities: the simulator manager, which enables mosaik to directly import a given simulator and execute it within the mosaik simulation process (if it is written in Python 3), to start a new simulator process and connect to it, and to interconnect to an already running simulation process.

The *Scheduler* is in control of the operation of the different simulators and manages the dataflow between them. In order to do this, it employs the event-discrete simulation library SimPy[5]. This framework is able to perform simulations in simulation time (as fast as possible). In addition, it allows mosaik to handle several simulators with possibly different and variable step sizes (as long as they use seconds as the time convention unit).

B. Classification

Based on the previously introduced classifications in Section II the scope of mosaik is defined as shown in Fig. 2. For the representation a morphological box has been chosen [34]. It depicts all attributes and their possible configurations. The attributes that count for mosaik are highlighted accordingly.

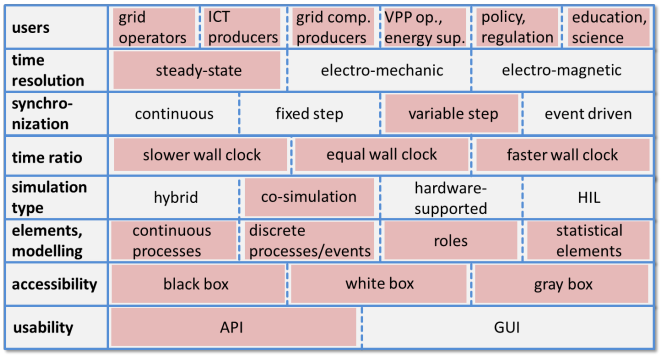

Fig. 2. Typing of mosaik using a morphological box

Mosaik addresses all user groups since it is a platform that can be used to functionally integrate simulators related to the particular use case. Furthermore, even though it is not limited to resolutions above 1 s, it is still most suitable for steady-state simulations. The scheduler used by mosaik is based on SimPy which makes it possible to operate with variable time steps. Integrating other schedulers such as Ptolemy[6] is indeed possible but would take considerable efforts. However, integrating Ptolemy is a task that is considered in the future development of mosaik. Mosaik can change the time ratio of the overall simulation specific to the considered use case. More precisely, the time ratio depends on the connected simulators that raise the requirements accordingly. Due to its nature being a co-simulation platform, it is possible to integrate all types of models regardless of their accessibility. Finally, mosaik provides an API and no GUI so far.

IV. Conclusion & Outlook

Appropriate simulations will be the key to handle the complexity in future energy systems. However, due to the many different systems involved, terms and concepts in the simulation-context are not distinctly defined. This often leads

[5]http://sourceforge.net/projects/simpy/
[6]http://ptolemy.eecs.berkeley.edu/

to misunderstandings in communication and development. In order to address this problem, a classification scheme for simulation approaches is suggested in this paper. It is based on various attributes each specified by a set of possible configurations. The goal of the proposed scheme is to support the dialog between developers and users of simulation technologies and help them to find suitable tools for their simulation purposes. The mosaik framework has been selected as a representative example to demonstrate the applicability of the scheme.

This paper is intended to serve as a base for discussion in the science community. After possibly necessary adjustments the next step to apply the scheme to today's established co-simulation approaches.

We have pointed out the issue of white box models becoming virtually "black" to users of interdisciplinary co-simulations, which makes it necessary to develop tools and automated processes to treat and analyze – potentially even optimize – co-simulation setups that are largely constructed from black box models. Even though the internal representations of such black box models can not be fully understood, this may be the only feasible option for future highly integrated and interdisciplinary energy systems. Efficiency and limitations of appropriate tools and methods will be investigated further.

REFERENCES

[1] P. Palensky and F. Kupzog, "Smart Grids," *Annual Reviews of Environment and Resources*, vol. 38, pp. 201–226, 11 2013. [Online]. Available: http://www.annualreviews.org/doi/abs/10.1146/annurev-environ-031312-102947

[2] J. Padulles, G. Ault, and J. McDonald, "An integrated SOFC plant dynamic model for power systems simulation," *Journal of Power Sources*, vol. 86, no. 1, pp. 495–500, 2000.

[3] A. K. Hartmann, *Practical guide to computer simulations*. World Scientific, 2009.

[4] E. de Jong, R. de Graff, P. Vassen, P. Crolla, A. Roscoe, F. Lefuss, G. Lauss, P. Kotsampopoulos, and F. Gafaro, "European white book on real-time power hardware in the loop testing: Derlab report no. r-005.0," 2012.

[5] K. Mets *et al.*, "Combining power and communication network simulation for cost-effective Smart Grid analysis," *IEEE Communications Surveys & Tutorials*, 2014.

[6] J. Bastian, C. Clauss, S. Wolf, and P. Schneider, "Master for co-simulation using FMI," in *8th International Modelica Conference, Dresden*, 2011.

[7] B. P. Zeigler, H. Praehofer, and T. G. Kim, *Theory of modeling and simulation: Integrating discrete event and continuous complex dynamic systems*. Academic press, 2000.

[8] I. Sommerville, "Integrated requirements engineering: A tutorial," *Software, IEEE*, vol. 22, no. 1, pp. 16–23, 2005.

[9] H.-J. Appelrath, H. Kagermann, C. Mayer *et al.*, *Future Energy Grid: Migrationspfade in das Internet der Energie*. Springer-Verlag, 2012.

[10] X. Fang, S. Misra, G. Xue, and D. Yang, "Smart Grid - The new and improved power grid: A survey," *Communications Surveys & Tutorials, IEEE*, vol. 14, no. 4, pp. 944–980, 2012.

[11] M. J. Gallivan and M. Keil, "The user – developer communication process: A critical case study," *Information Systems Journal*, vol. 13, no. 1, pp. 37–68, 2003.

[12] M. Schreiber and P. Hochloff, "Capacity-dependent tariffs and residential energy management for photovoltaic storage systems," in *IEEE Power and Energy Society General Meeting, Vancouver*, 2013.

[13] J. M. Carrasco, L. G. Franquelo, J. T. Bialasiewicz, E. Galván, R. P. Guisado, M. A. Prats, J. I. León, and N. Moreno-Alfonso, "Power-electronic systems for the grid integration of renewable energy sources: A survey," *Industrial Electronics, IEEE Transactions on*, vol. 53, no. 4, pp. 1002–1016, 2006.

[14] S. Rohjans, C. Dänekas, and M. Uslar, "Requirements for smart grid ICT-architectures," in *Innovative Smart Grid Technologies (ISGT Europe), 2012 3rd IEEE PES International Conference and Exhibition on*. IEEE, 2012, pp. 1–8.

[15] A. Niese, S. Lehnhoff, M. Tröschel, M. Uslar, C. Wissing, H. Appelrath, and M. Sonnenschein, "Market-based self-organized provision of active power and ancillary services: An agent-based approach for smart distribution grids," in *Complexity in Engineering (COMPENG), 2012*. IEEE, 2012, pp. 1–5.

[16] Dena-Netzstudie II, "Integration erneuerbarer Energien in die deutsche Stromversorgung im Zeitraum 2015 – 2020 mit Ausblick 2025," *Berlin, Germany, Nov*, 2010.

[17] F. Milano, "An open source power system analysis toolbox," *Power Systems, IEEE Transactions on*, vol. 20, no. 3, pp. 1199–1206, 2005.

[18] T. Strasser, M. Stifter, F. Andren, and P. Palensky, "Co-simulation training platform for smart grids," *Power Systems, IEEE Transactions on*, 2014.

[19] P. Palensky, E. Widl, and A. Elsheikh, "Simulating cyber-physical energy systems: challenges, tools and methods," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 44, no. 3, pp. 318–326, 2013.

[20] F. Milano, *Power system modelling and scripting*. Springer-Verlag Berlin Heidelberg, 2010.

[21] J. T. Buck, S. Ha, E. A. Lee, and D. G. Messerschmitt, "Ptolemy: A framework for simulating and prototyping heterogeneous systems," 1994.

[22] S. Yoo and K. Choi, "Optimistic timed HW - SW co-simulation," in *in Proc. of APCHDL97*. Citeseer, 1997.

[23] A. M. Kosek, O. Lünsdorf, S. Scherfke, O. Gehrke, and S. Rohjans, "Evaluation of Smart grid control strategies in co-simulation – Integration of IPSYS and mosaik," in *in 18th Power Systems Computation Conference (PSCC2014)*, 2014.

[24] S. C. Glotzer, S. Kim, P. T. Cummings, A. Deshmukh, M. Head-Gordon, G. Karniadakis, L. Petzold, C. Sagui, and M. Shinozuka, "International assessment of research and development in simulation-based engineering and science. Panel report," DTIC Document, Tech. Rep., 2009.

[25] D. Van Beek and J. Rooda, "Languages and applications in hybrid modelling and simulation: Positioning of Chi," *Control Engineering Practice*, vol. 8, no. 1, pp. 81–91, 2000.

[26] P. Palensky, E. Widl, A. Elsheikh, and M. Stifter, "Modeling intelligent energy systems: Co-simulation platform for validating flexible-demand EV charging management," *IEEE Transactions on Smart Grids*, vol. 4, no. 4, pp. 1939–1947, 12 2013.

[27] M. E. Khan and F. Khan, "A comparative study of white box, black box and grey box testing techniques," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 3, no. 6, 2012.

[28] N. R. Kristensen, H. Madsen, and S. B. Jørgensen, "A method for systematic improvement of stochastic grey-box models," *Computers & chemical engineering*, vol. 28, no. 8, pp. 1431–1449, 2004.

[29] B. Myers, S. E. Hudson, and R. Pausch, "Past, present, and future of user interface software tools," *ACM Trans. Comput.-Hum. Interact.*, vol. 7, no. 1, Mar. 2000.

[30] J. Bloch, *Effective Java* - , 2nd ed. Boston: Addison-Wesley Professional, 2008.

[31] S. Schütte, S. Scherfke, and M. Tröschel, "Mosaik: A framework for modular simulation of active components in Smart Grids," in *2011 IEEE First International Workshop on Smart Grid Modeling and Simulation (SGMS)*, 2011, pp. 55–60.

[32] S. Rohjans, S. Lehnhoff, S. Schütte, S. Scherfke, and S. Hussain, "Mosaik – a modular platform for the evaluation of agent-based Smart Grid control," in 4^{th} *IEEE PES International Conference and Exhibition on Innovative Smart Grid Technologies (ISGT Europe)*, 2013.

[33] M. Buescher, A. Claassen, M. Kube, S. Lehnhoff, K. Piech, S. Rohjans, S. Scherfke, C. Steinbrink, J. Velasquez, F. Tempez, and Y. Bouzid, "Integrated Smart Grid simulations for generic automation architectures with RT-LAB and mosaik," in 5^{th} *IEEE International Conference on Smart Grid Communications*, 2014.

[34] F. Zwicky, "Discovery, invention, research through the morphological approach," 1969.