Towards a Tool-Based Approach for Dynamically Generating Co-Simulation Scenarios based on complex Smart Grid System Architectures

Christoph Binder
and Michael Fischinger
and Christian Neureiter
and Goran Lastro

Center for Secure Energy Informatics
Urstein Sued 1, A–5412 Puch/Salzburg, Austria
Email: {firstname.lastname}@fh-salzburg.ac.at

Katharina Polanec
and Jounes-Alexander Gross

Salzburg University of Applied Sciences
Urstein Sued 1, A–5412 Puch/Salzburg, Austria
Email: kpolanec.itsb-m2019@fh-salzburg.ac.at
jgross.its-m2018@fh-salzburg.ac.at

***Abstract*—The advancement of the electricity system in the upcoming years faces major challenges due to the rising number of decentralized and self-managed participants that individually react to the current load. These participants, regardless whether being an electric Vehicle or a Photovoltaic System, can have diverse behaviors, whose interplaying one mutual system is still widely unexplored. As conventional engineering methods reach their limits when trying to predict the collective behaviors of such Systems of Systems (SoS), the utilization of advanced tools such as the SGAM Toolbox and Mosaik framework is necessary. Each of the tools provide a distinct functionality in their unique field of application, however, their interconnection has yet to be introduced. Therefore, this work has been dedicated to explore the possibility of extending an SGAM based model so that the generation of components and their behavior for usage in the Co-Simulation environment of Mosaik is supported. The developed artifacts are thereby evaluated with a suitable real-world case study making use of different kinds of Electric Vehicle (EV) behaviors. Based on the results of this approach, entire Co-Simulation scenarios can be set up according to previously modeled Smart Grid architectures, which enables the analysis of different system behaviors in a considerably simplified way.**

***Index Terms*—Co-Simulation, Emergent Behavior, Electric Vehicles, Smart Grid Architecture Model (SGAM), Domain-specific Systems Engineering (DSSE)**

I. Introduction

Advances in research and development lead to constantly increasing possibilities utilized by the Internet of Things (IoT) and its components, so-called Cyber-physical Systems (CPSs). This trend leads to an integration of more and more intelligent devices into the current power system, which slowly results into the transition of the traditional unidirectional energy flow into a dynamical network containing multiple producers and consumers. Thus, the term Smart Grid is increasingly gaining in importance in recent years, although it is a topic to be discussed for over a decade now. One effect concluding from this trend is the rising popularity of e-mobility. Just to mention an example, a decentralized Electric Vehicle (EV) charging management system improving the reliability and sustainability of electricity production or distribution by utilizing the benefits of IoT technologies is introduced in [1]. Taking the interplay of these self-monitoring and dynamically adapting Smart Grid components into further consideration, to manage the energy balance of the complete power system becomes a difficult challenge to be addressed. This is furthermore underpinned by the Smart Grid being a complex System of Systems (SoS), which can be attributed to the large number of components and their dynamic behavior, as explained in [2]. Additionally, it is shown that the effects resulting from the interplay of multiple EVs in a power system are mostly unpredictable and usually undesirable, leading to the topic of emergent behavior. Thus, nowadays systems engineering needs to consider this issue with the aim of finding countermeasures to these behaviors before implementing the system.

Taking this into further consideration, concepts introduced by Model-Based Systems Engineering (MBSE) seem to provide a solution for dealing with the complexity in future energy systems. Several approaches targeting the modeling of such a system and providing the possibility to simulate its effects have been published in recent years, just to mention a few [3]–[6]. However, although all mentioned publications introduce model-based engineering of modern power systems, the proposed main objective differs between the single approaches. While some of them are focused on modeling of specific stakeholder concerns, others achieve to integrate common standards. As MBSE has become more popular for developing Smart Grid applications, approaches for simulating them needed to arise. Thus, the authors of this paper already proposed an approach resulting from new researches introducing an interface between the architectural model and the Co-Simulation scenario, where the previously modeled behavior can be exported as a simulator for the simulation environment with just a few clicks [7].

However, as the importance of this functionality has been pointed out, one big missing issue has been mentioned in this and other publications. While a Co-Simulation scenario usually is composed of several simulators of single components, a Smart Grid model does not provide the possibility to inherit cardinality information. This means, a use case of one EV is not different from one containing 100 EVs. More precisely, the current version of the framework solely allows to create one simulator based on one specific model, which exhibits two problems. First, in a large-scale system like a Smart Grid is, probably single components implement different behaviors. Taking the example of EVs, not each single car follows the same charging strategy. Second, as it is difficult to model 100 different EVs in Smart Grid Architecture Model (SGAM), but a co-simulation needs to instantiate each EV on its own and with its respective functionality, this has to be considered when setting up a dynamic simulation scenario. Therefore, this paper introduces a way for adding this cardinality information into a Smart Grid model and utilizing it for preparing the simulation environment. Furthermore, since the number of components is crucial when trying to explore the effects of such a system, the developed interface is extended to consider the information of each individual model and export it to the corresponding co-simulation scenario. By doing so, the SGAM Toolbox is utilized for modeling the system, whose Domain Specific Language (DSL) is adapted at first, and the Mosaik framework is used for executing the simulation. As evaluation, a comprehensive real-world case study is applied to verify the intended functionality of the proposed approach.

To address these aspects, this contribution is structured as following: In Section II an overview of SGAM, the Mosaik framework regarded to co-simulation and the toolchain of Domain Specific Systems Engineering (DSSE) is given. Hereafter, the used research methodology and the description of the case study is stated in Section III. The implementation itself is summarized in Section IV Based on the previously defined use case, the applicability is demonstrated in Section V. Finally, in Section VI the results of the conducted study are summarized and then a conclusion is given.

II. Related Work

A. Smart Grid Architecture Model (SGAM)

The Smart Grid Architecture Model (SGAM), depicted in Figure 1, is the result of standardization efforts initiated by the Smart Grid Coordination Group (SGCG), a joint project driven by the European standardization bodies European Telecommunications Standardization Institute (ETSI) and European Committee for Electrotechnical Standardization (CENELEC). The main outcome of this work was the formalization of the three-dimensional architecture model, capable of providing a standardized composition of Smart Grid systems with the purpose to generate a holistic view, as explained in the context of the Standardization Mandate M/490 [8]. Originating from the well-known NIST Domain Model [9], a major part of the SGAM falls back on its specifications.

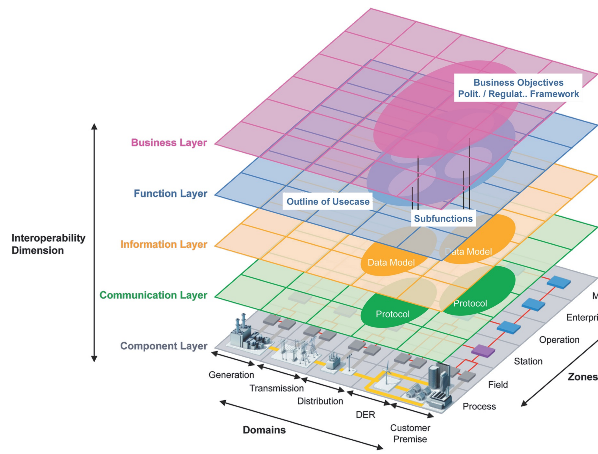

Fig. 1: The Smart Grid Architecture Model (SGAM) [8]

Thus, the domain-axis of the model decomposes an electric energy system and its interrelations focusing on either electrical connections as well as logical interactions. By doing so, the following domains have been specified: *Bulk Generation* representing the generation of electrical energy in bulk quantities, *Transmission* and *Distribution* for its transportation as well as *DER* or *Customer Premises* representing energy consumers. On the other hand, the domain model has been extended and combined with the SCADA automation pyramid in order to integrate Information and Communication Technology (ICT) components with their functionality according to automation possibilities. As the thereby resulting plane solely provides guidance for decomposing and structuring a Smart Grid based system, a third axis providing five layers derived from the GWAC Interoperability Stack has been introduced. Thus, the layers of the SGAM are defined as Business, Function, Information, Communication and Component Layer.

B. Model-based Systems Engineering (MBSE)

As mentioned earlier, modeling in the Smart Grid and simulation of its components during runtime is no unknown territory. During the last decade, similar approaches have been in the focus of several research projects. As an example, first attempts of utilizing MBSE to model a power system from a SoS perspective in order to manage its complexity and address the concerns of different stakeholders have been proposed in [10]. Although being in early stages of research, some introduced concepts still find usage at current times, like using planes for structuring a Smart Grid or applying the Systems Modeling Language (SysML) for modeling the system. Additionally, in [4] a methodology specially focused on modeling the behavior of CPS in the Smart Grid is introduced. By simulating EVs, first effects of their individual behaviors towards the complete power system could be investigated. Moreover, as standardization in the Smart Grid became a more popular topic, approaches making use of the Common Information Model (CIM), the IEC 61850 or the IEC 61499 emerged, like the framework proposed in [5]. Furthermore, the author of [11] also introduced the designing of EVs with regard to their autonomy by focusing on a rechargeable battery with solar panels on the vehicle's roof.

A more complete approach inheriting an underlying architectural framework and introducing an adequate development process has been introduced in [12]. This methodology remains in constant development and resulted in the proposal of DSSE, which is observable by the continuously updated SGAM Toolbox. The applicable set of tools inherits supportive methods for consistent system development in different phases of the development life-cycle. Thus, to finally reach a SoS "Integration Toolchain" for Smart Grids, the DSSE approach is split into eight single steps that each contribute to the superordinate goal of creating holistic Smart Grid applications on their own way and with chosen methods.

C. Co-Simulation

Simulating complex systems in the Smart Grid is not a completely new topic to discuss about. Therefore, a collection of suitable tools and methods has been published by [13]. However, with most frameworks focusing on simulating the behavior of single components and the Smart Grid inheriting SoS characteristics, the need for Co-Simulation methodologies has become more obvious in order simulate an energy system on system-level. Thus, first attempts have been proposed by [14], where the authors simulate Photovoltaic Cells (PCs) and try to find countermeasures for dealing with a temporary loss of power. A more generic example has been introduced in [15], where a framework for event-driven simulation of previously modeled Smart Grid components is provided. Moreover, in [16] a simulation approach intelligently controlling the energy demand based on fuzzy logic and considering aspects like electricity prices or renewable production is illustrated.

Nevertheless, under the term of DSSE, a more comprehensive approach is introduced, which targets the whole lifecycle of systems engineering, introduces a complete toolchain and provides adequate tool-support, based on the SGAM. Taking the seventh step of DSSE into further consideration, a specific framework has been chosen to work best with SGAM based models. Thus, due to its applicability and its dynamic adaptability, Mosaik has been the tool of choice, which is mainly substantiated by its thoroughly tailoring to the Smart Grid. Furthermore, the open source tool is written in Python and integrates a specific power grid simulator like PyPower. This means, that existing Smart Grid models can be used and instantiated in order to create simulators for large-scale simulation scenarios [17]. By doing so, Mosaik exists of four different main components that interact with each other to ensure optimized outcome. First, the *Simulator-API* represents each component of the simulation scenario and its behavior, while the *Scenario-API* deals with processing the simulation from its initialization to connecting the simulator during its runtime. Moreover, in order to ensure a controlled interconnection of simulators, the *Simulator-Manager* as well as the so-called *Coordination Execution* have been implemented [18]. The interplay of the mentioned components are composed to a framework able to simulate the behavior of EVs seen as single systems themselves or in a SoS context. By doing so, a typical characteristic defining a Co-Simulation is addressed, as proposed in [19]. On the one hand, the respective simulators are coupled individually with each other, on the other hand a generic coupling utilizing any kind of middleware is enabled by the management components. Thus, simulators can either communicate directly or via a central unit dealing with data exchange or time synchronization [20].

III. Approach

As already mentioned, the main objective of this approach is to enable the possibility for adding cardinality information into models based on SGAM. Furthermore, this information is exported from the model and used for setting up the Co-Simulation scenario. More precisely, this means that all components in the Smart Grid need to be filtered from the model and the same number of simulators is created, one for each component. By doing so, each simulator contains the behavior of the Smart Grid elements, which is modeled in detail in the according Function Layer of SGAM. Subsequently, the simulators, respectively their behavior, is utilized by Mosaik in order to simulate the previously modeled system. As the exact outcome of this approach cannot be foreseen due to many influencing factors as well as dynamically changing environmental aspects, an agile development method needs to be applied. Thus, the contributions of the Agile Design Science Research Methodology (ADSRM) seem to be suitable for application for addressing the mentioned issues. However, as the chosen entry point is the utilization of a particular case study in this specific project, the first step is to define an actual Smart Grid application scenario, which is described in more detail in the following.

A. Case Study Design & Requirements

As the goal is to simulate a complete model based on SGAM, a suitable use case needs to be chosen. With respect to this, a fitting case study has already been defined in one of the previous iterations of ADSRM, as described in [2]. Thus, the existing concepts can be taken for use and adapted to the particular scenario proposed in this contribution. This means, the number of 20 households and 20 connected EVs controlled by variable charging algorithms remains as previously specified. However, a specific feature of the ajusted case study is the classification of the EVs into four different clusters, each one containing three respectively seven households. This clustering allows the usage of different kinds of variable charging algorithms for the EVs and is depicted in Figure 2. The resulting characteristic is important to investigate different kind of charging behaviors and their effects on the overall power system. However, as this would exceed the scope of this work, only the export of each single behavior into a simulator and its application will be considered by this approach. By doing so, new requirements have been defined for the current iteration of ADSRM again by using methods described in [21]:

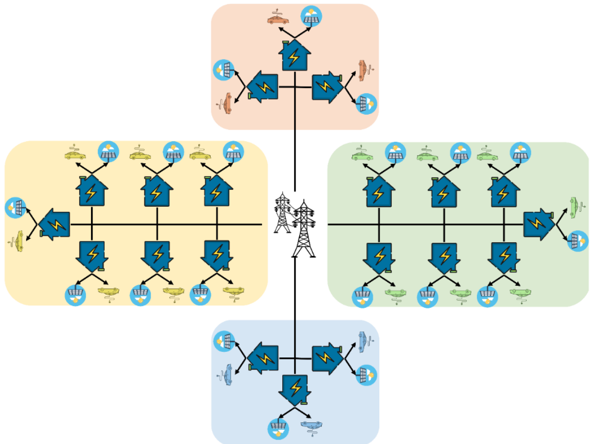

Fig. 2: Case Study used by ADSRM [22]

1) The system should provide the possibility to create a fully comprehensive Smart Grid model. Thereby, each involved component and its behavior has to be considered. Hence, a comprehensive model of the case study's system architecture including all EVs, photovoltaic cells and smart home components needs to be created first.
2) The system should simulate the previously created model as detailed as possible. Thus, all information from the model needs to be utilized for their application in Mosaik. This means, the DSL and the corresponding interface between the two tools have to be adapted.

IV. Implementation

The first step enabling the possibility of adding cardinality into an SGAM based model is to refine the DSL of the SGAM Toolbox. Since this language offers several domain-specific elements, characteristic features and wide-ranging functionalities to create such a model of an electrical system, it has already been used for describing complex models of international energy projects. Thus, providing an interface to a Co-Simulation framework would furthermore enhance the usability of the toolbox. Based on the changes, the corresponding interface creating all files needed for the simulation scenario can be adapted.

A. Adaption of the DSL

As already mentioned, considerations about adapting the DSL need to be elaborated at first. There are two possibilities of adding cardinality information into an SGAM model. The first way is to individually create each single element and connect it to the power grid. By doing so, a dedicated artifact inheriting all needed simulator values has been added. Examples for these values are in- and outputs of the simulator and the step size, in which the simulator is executed. Furthermore, each of the modeled elements is described in more detail with an appropriate behavior, which is located in the Function Layer. For this purpose, activity diagrams are utilized, since they offer the possibility of model-based code generation. However, creating each single element of the model is a time-consuming task including several repetitions, which is only suitable for smaller systems including a manageable number of elements. For more complex systems, another way of considering all elements has been provided.

In more detail, Smart Grid components with similar behaviors and interfaces to the power grid, like photovoltaic cells or EVs, can be summarized to one element in the model, which contains information about the behavior of the component, simulator configurations and the actual number of this kind of element within the system. It can be seen as a class-instance relation, where the class is modeled and according to each quantity a simulator instance is created. This allows clear and structured modeling of extensive systems on the one hand, but detailed simulation possibilities on the other hand.

B. Simulation Scenario Generation

Based on the previously mentioned aspects, a suitable simulation environment needs to be generated. In order to enable this, the interface setting up the scenario file and the single simulators needs to be adapted to the changes in the DSL. This means, for each element in the model, a simulator file is created. Depending on whether each single element or the summarized components are modeled, the behavior of each element is exported into JAVA-code, which can be executed from the Co-Simulation.

A novelty of this approach is the generation of the complete main scenario file of Mosaik. This Python-file amongst others deals with setting up the simulation environment by instantiating all simulators and connecting them to each other as well as to the grid. This is furthermore supported by initiating the database and the web visualization for observational purposes. Technically speaking, the python code of the main scenario file is saved as a template in the SGAM Toolbox, where the parts concerning the initialization of the simulators as instances and their interconnection can be processed during runtime according to the underlying Smart Grid model. As this file is essential for controlling the Co-Simulation process, dynamically generating it, based on the corresponding model, improves the possible application areas for the link between the architectural model and the simulation framework. For example, setting up a Co-Simulation scenario based on a previously modeled SoS with just a few clicks addresses aspects like investigating different kinds of emergent behavior or evaluating the system before its implementation.

V. Application

To evaluate the functionality of the presented approach, the previously defined case study[1] is utilized for its modeling and simulation. As mentioned before, a number of twenty households operating EVs as well as photovoltaic cells are considered to be a suitable use case in this application scenario. By doing so, this specific scenario consists of three steps. First a suitable model is generated according to the

[1] https://sgam-toolbox.org/Co-Simulation

specifications of SGAM. Subsequently, the information of the model is exported and used for setting up the Co-Simulation, which is executed in the last step. Thus, in the following the creation of the model is described. As providing an overview of the complete model would exceed the scope of this work, only specific aspects are considered in more detail.

According to these considerations, an important part of the model is to develop the SGAM Component Layer. This viewpoint provides all technical information and a detailed overview of the single components in the grid. According to the horizontal axis of SGAM, those components are aligned from generating over transmitting up to consuming the electrical energy. However, since the photovoltaics are directly attached to the households, most of the elements, regardless of whether producing or consuming energy, are located in the customer premises pane, as visualized in Figure 3.

As depicted in the figure, the two different possibilities of modeling the single elements have been utilized. At the top three sub-grids, every single element has been modeled according to its position in the grid. By doing so, the interfaces and the data exchange is modeled on the Communication respectively the Information Layer of SGAM. Furthermore, each of the modeled EV is described in more detail with an appropriate charging behavior. In this example, four different kinds of algorithms have been used, which are located in the SGAM Function Layer. More precisely, one behavior only allows the EV to be charged beyond a certain price. Thus, an individual price cap is added to each simulator, which reacts to the current electricity price within the grid. Another algorithm restricts the charging process to the photovoltaic cells. If the cells attached to each household produce energy, the connected EV is allowed to be charged. The two remaining behaviors use more simple algorithms, one describes a queue, where the vehicles entering first are allowed to be charged first, the other one delineates a parallel charging process, where each EV is allowed to be charged consecutively for one hour at most. Consequently, the two of the mentioned charging behaviors and their assignment to the EV can be seen in Figure 4.

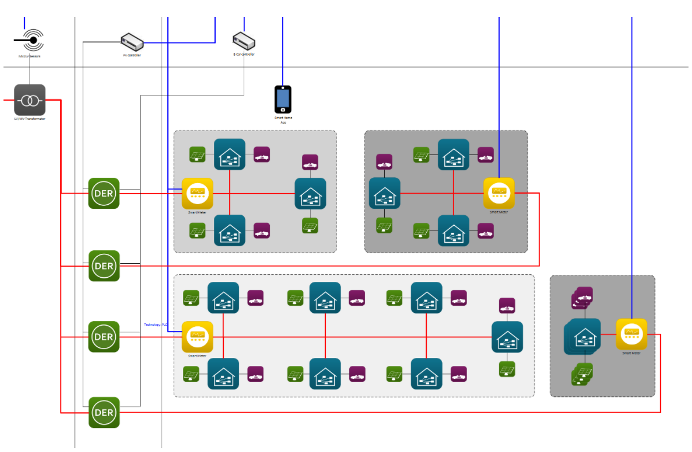

Fig. 3: SGAM Component Layer

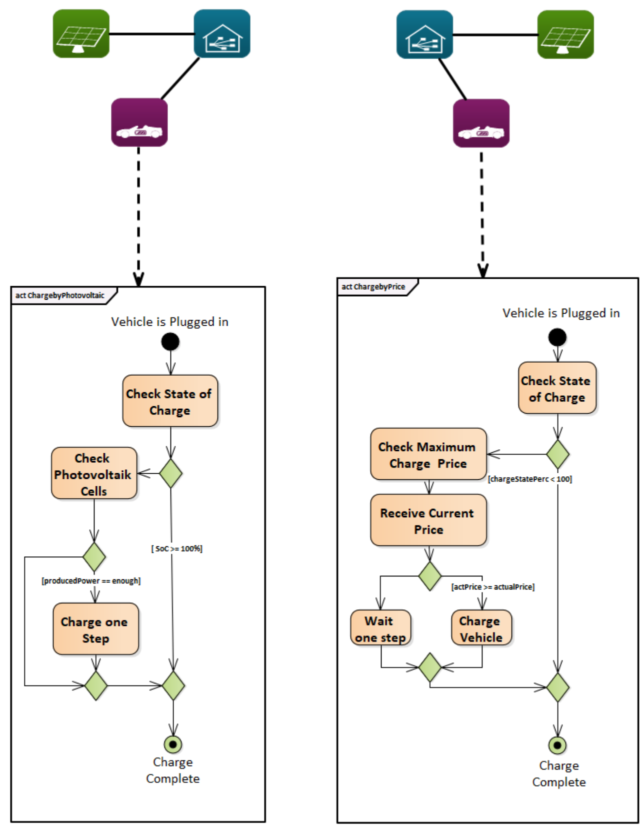

Fig. 4: SGAM Component Layer

After modeling the case study and setting up the attributes needed for the interface in order to create the simulators and the main scenario file, this step can be executed. This is done by calling the according functionality of the SGAM Toolbox. By doing so, the charging behaviors are translated into code, which is then embedded to the respective simulator. Moreover, all simulators and the dynamically generated main file are copied into their place in the folder structure of Mosaik. This allows the execution of the Co-Simulation in Mosaik with just one click, as everything has been prepared to optimize the usability. To complete this, by executing the simulation of this case study, the overall electricity consumption depicted in Figure 5 appears. However, as the main objective of this approach is just to set up the Co-Simulation scenario based on the modeled system and not to optimize the balance between production and consumption, those values are not further interpreted.

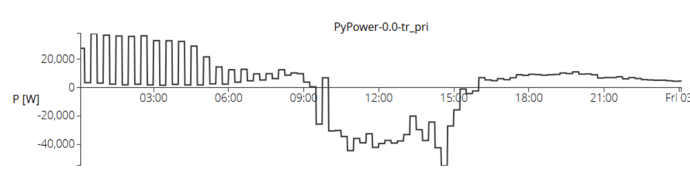

Fig. 5: Mosaik overall consumption

VI. Conclusion & Future Work

The presented work introduces a new way for modeling and simulating complex and extensive Smart Grid systems, by providing users the possibility to dynamically create the Co-Simulation environment based on a previously modeled architecture with just a few clicks. This paves the way for future projects in this area on the one hand and enhances the usability of this interface on the other hand. In order to do so, the approach has been split up in three steps. First, a suitable case study of a small excerpt of a power grid is presented in Section III. After defining the requirements, the DSL and the export functionality of the SGAM Toolbox has been extended with the goal to create the main file of Mosaik, which controls the whole simulation environment. Detailed information about this is given in Section IV. In the last step, the whole simulation scenario is modeled according to SGAM before exporting its information and setting up the simulation environment by creating all simulators and interconnecting them in the main file. Subsequently, the whole scenario is executed with Mosaik, as seen in Section V. Comparing the result of this paper to the previously developed interface [7], it is now possible to model a Smart Grid system including all contained components and simulate the whole scenario by automatically setting up the main scenario file as well as creating all interconnections with the created simulators.

The outcome of this work paves the way for further projects targeting the simulation Smart Grid systems. By improving the interface between modeling und simulating power systems the workload to interconnect the single tools has been considerably reduced. This means, simulating and in further consequence evaluating or interpreting the behavior of such complex systems including a lot of components has been facilitated. Based on this, a new project could deal with demonstrating different kinds of emergent behavior in a more sophisticated and large-scale SoS, in order to apply demand side management techniques to counteract with these unpredictable and undesirable behaviors before implementing the system. Furthermore, to consider all particularities of such a SoS, the overlapping of the Smart Grid domain and other domains like Automotive or Smart City has to be considered. However, such a cross-domain approach is still in an infancy stage, but its gaining in importance due to the continuous merging of these areas.

Acknowledgment

The support for valuable contributions of LieberLieber Software GmbH and successfactory consulting group is gratefully acknowledged. The financial support by the Federal State of Salzburg is also gratefully acknowledged.

References

[1] L. Yao, Y.-Q. Chen, and W. H. Lim, "Internet of things for electric vehicle: An improved decentralized charging scheme," in *2015 IEEE International Conference on Data Science and Data Intensive Systems (DSDIS)*. IEEE, 2015, pp. 651–658.

[2] C. Binder, J.-A. Gross, C. Neureiter, and G. Lastro, "Investigating emergent behavior caused by electric vehicles in the smart grid using Co-Simulation," in *2019 14th Annual Conference System of Systems Engineering (SoSE)*. Anchorage, Alaska, USA: IEEE, 2019.

[3] A. J. Lopes, R. Lezama, and R. Pineda, "Model based systems engineering for smart grids as systems of systems," *Procedia Computer Science*, vol. 6, pp. 441–450, 2011.

[4] I. Lampropoulos, G. M. Vanalme, and W. L. Kling, "A methodology for modeling the behavior of electricity prosumers within the smart grid," in *2010 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT Europe)*. IEEE, 2010, pp. 1–8.

[5] F. Andrén, M. Stifter, and T. Strasser, "Towards a semantic driven framework for smart grid applications: Model-driven development using cim, iec 61850 and iec 61499," *Informatik-Spektrum*, vol. 36, no. 1, pp. 58–68, 2013.

[6] C. Neureiter, M. Uslar, D. Engel, and G. Lastro, "A standards-based approach for domain specific modelling of smart grid system architectures," in *Proceedings of International Conference on System of Systems Engineering (SoSE) 2016*, Kongsberg, Norway, 2016, pp. 1–6.

[7] C. Binder, M. Fischinger, L. Altenhuber, D. Draxler, C. Neureiter, and G. Lastro, "Enabling architecture based co-simulation of complex smart grid applications," in *The 8th DACH+ Conference on Energy Informatics*. Salzburg, Austria: Springer, 2019.

[8] CEN-CENELEC-ETSI Smart Grid Coordination Group, "Smart Grid Reference Architecture," 2012.

[9] C. Greer, D. A. Wollman, D. E. Prochaska, P. A. Boynton, J. A. Mazer, C. T. Nguyen, G. J. FitzPatrick, T. L. Nelson, G. H. Koepke, A. R. Hefner Jr *et al.*, "Nist framework and roadmap for smart grid interoperability standards, release 3.0," Tech. Rep., 2014.

[10] A. J. Lopes, R. Lezama, and R. Pineda, "Model Based Systems Engineering for Smart Grids as Systems of Systems," *Procedia Computer Science*, vol. 6, pp. 441–450, 2011.

[11] K. Tehrani and O. Maurice, "A cyber physical energy system design (cpesd) for electric vehicle applications," in *2017 12th System of Systems Engineering Conference (SoSE)*. IEEE, 2017, pp. 1–6.

[12] C. Neureiter, *A Domain-Specific, Model Driven Engineering Approach for Systems Engineering in the Smart Grid*, Hamburg, Germany, 2017.

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

[14] T. Godfrey, S. Mullen, D. W. Griffith, N. Golmie, R. C. Dugan, and C. Rodine, "Modeling smart grid applications with co-simulation," in *2010 first IEEE International conference on smart grid communications*. IEEE, 2010, pp. 291–296.

[15] C.-h. Yang, G. Zhabelova, C.-W. Yang, and V. Vyatkin, "Cosimulation environment for event-driven distributed controls of smart grid," *IEEE Transactions on Industrial Informatics*, vol. 9, pp. 1423–1435, 2013.

[16] L. Roiné, K. Therani, Y. S. Manjili, and M. Jamshidi, "Microgrid energy management system using fuzzy logic control," in *2014 World Automation Congress (WAC)*. IEEE, 2014, pp. 462–467.

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

[18] C. Steinbrink, S. Lehnhoff, S. Rohjans, T. I. Strasser, E. Widl, C. Moyo, G. Lauss, F. Lehfuss, M. Faschang, P. Palensky *et al.*, "Simulation-based validation of smart grids–status quo and future research trends," in *International Conference on Industrial Applications of Holonic and Multi-Agent Systems*. Springer, 2017, pp. 171–185.

[19] P. Palensky, A. A. Van Der Meer, C. D. Lopez, A. Joseph, and K. Pan, "Cosimulation of intelligent power systems: Fundamentals, software architecture, numerics, and coupling," *IEEE Industrial Electronics Magazine*, vol. 11, no. 1, pp. 34–50, 2017.

[20] F. Schloegl, S. Rohjans, S. Lehnhoff, J. Velasquez, C. Steinbrink, and P. Palensky, "Towards a classification scheme for co-simulation approaches in energy systems," in *2015 International Symposium on Smart Electric Distribution Systems and Technologies (EDST)*. IEEE, 2015, pp. 516–521.

[21] K. Pohl and C. Rupp, *Basiswissen Requirements Engineering: Aus- und Weiterbildung nach IREB-Standard zum Certified Professional for Requirements Engineering Foundation Level*. dpunkt. verlag, 2015.

[22] K. Conboy, R. Gleasure, and E. Cullina, "Agile Design Science Research," in *New Horizons in Design Science: Broadening the Research Agenda*. Cham: Springer International Publishing, 2015, pp. 168–180.