Investigating Emergent Behavior caused by Electric Vehicles in the Smart Grid using Co-Simulation

Christoph Binder
Center for Secure Energy Informatics
Salzburg University of Applied Sciences
Urstein Sued 1, A5412 Puch/Salzburg, Austria
christoph.binder@fh-salzburg.ac.at

Jounes-Alexander Gross
Center for Secure Energy Informatics
Salzburg University of Applied Sciences
Urstein Sued 1, A5412 Puch/Salzburg, Austria
jgross.its-m2018@fh-salzburg.ac.at

Christian Neureiter
Center for Secure Energy Informatics
Salzburg University of Applied Sciences
Urstein Sued 1, A5412 Puch/Salzburg, Austria
christian.neureiter@fh-salzburg.ac.at

Goran Lastro
Center for Secure Energy Informatics
Salzburg University of Applied Sciences
Urstein Sued 1, A5412 Puch/Salzburg, Austria
goran.lastro@fh-salzburg.ac.at

***Abstract*—Today's electricity system faces major challenges by the ongoing integration of decentralized, renewable energy resources and individual participants benefitting from the Internet of Things (IoT), like Electric Vehicles (EVs) or Smart Meters. The interplay of these autonomous components forms the popular term of the so-called Smart Grid. Since social mannerism may result in simultaneous charging cycles of EVs in such a System of Systems (SoS), ominous peak loads are expected to emerge. Thus, to deal with this often unpredictable behavior before implementing the system, usually a simulation is applied. Therefore, this paper proposes a co-simulation approach using Mosaik, a framework tailored to the Smart Grid domain. By doing so, the power system including several EVs and their charging strategy is modeled according to the Smart Grid Architecture Model (SGAM) in the first step. Next, in order to simulate and validate the system's emergent behavior, an excerpt of a real-world case study is utilized. Based on the outcome of this co-simulation, the practical investigation of Smart Grids can be improved by applying protruded demand side response approaches.**

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

I. Introduction

The transition of the original power grid towards the so-called Smart Grid is becoming increasingly apparent in recent years. This is mainly encouraged by the expected depletion of primary energy resources, more specific the diminishing of fossil fuels like crude oil or gas, according to [1], [2]. Additionally, technological advances promoted by the Internet of Things (IoT) offer new possibilities for the efficient use of more sustainable and environmentally-friendly resources. This leads to major changes affecting the unidirectional energy flow from centralized facilities towards its customers resulting in a dynamical network containing multiple producers and consumers. One effect concluding from this trend is the rising popularity of e-mobility. Expected to become a major means of transportation within the next few years [3], new possibilities for efficient energy-use are applied by those electric vehicles (EVs). For example, in [4] a decentralized EV charging management system is introduced, which improves reliability and sustainability of electricity production or distribution by exploiting the benefits of IoT technologies. Thus, these decentralized approaches and new strategies for distributing the load through demand side management and response techniques [5], [6] lead to a constantly increasing complexity in the Smart Grid. Therefore, new methods for developing and evaluating vehicular systems as part of current and future energy systems need to be defined.

As aforementioned, various individual challenges need to be addressed for realizing e-mobility within the Smart Grid. Hence, considering the composition of several EVs as an interdisciplinary system itself, the classification scheme proposed in [7] can be used for abstracting its complexity. According to this scheme, modern vehicles can be considered as complex systems. This is attributed to the large number of vehicles in a modern power system as well as their dynamic charging behavior. Additionally, falling back on the criteria mentioned in [8] and the Smart Grid being a system itself, the term System of Systems (SoS) is suggested to be used in order to emphasize the autonomous character of the system's individual participants. The use of this definition is substantiated by taking two real-world scenarios into consideration. Firstly, in [9], the difficulties that have to be addressed when elaborating interrelations between several application domains (ADs) within a Smart City are explained in more detail. In the proposed work, the typical characteristics of SoS *independent operation* and *geographic distribution* of its components are demonstrated by using the example of traffic management. The second example attends to the charging strategy of EVs according to electricity prices. Price fluctuations cause multiple vehicles to charge at the same time, which would be representative for promoting the possibility of unpredictable behaviors, as it is typical for a SoS.

Taking this into further consideration, many vehicles actively reacting to most of the same external influences such as the price of electricity leads to the subject of demand side. However, with several objects responding to the same factors on the demand side, a behavior that effects the environment contradictory to what is desired becomes observable, the so-called emergent behavior. Since this kind of effect is mostly unpredictable and usually undesirable, countermeasures need to be applied before implementing the system. Therefore, a co-simulation is considered being an appropriate measure to be able to realistically analyze the behavior of components within the Smart Grid. Tailored to this specific case, the framework Mosaik has been developed by OFFIS [10], which offers great connectability to the Smart Grid Architecture Model (SGAM) [11]. To consolidate this, several publications introduce the development of architectural models of energy systems aligned to the specifications of the SGAM [12], [13]. However, although there are comprehensive models of Smart Grid systems existing, the integration of an EV into such a model has not been proposed yet. Therefore, this paper deals with two major issues. At first a suitable case study for observing emergent behavior in the Smart Grid is selected and the suitability of SGAM for developing the architecture of such a vehicular-focused energy system is evaluated. Secondly, the resulting model is used in order to consider the appearance of emergent behavior within the architectural system.

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 several methods for developing Smart Grid architectures is given. Hereafter, the creation of the co-simulation itself is stated in Section III. Based on a suitable use case, the applicability is demonstrated in Section IV. Finally, in Section V the results of the conducted study are summarized and then a conclusion is given.

II. Related Work

A. Domain Specific Architecture Framework

In the context of the Standardization Mandate M/490, introduced by the European Commission, the Smart Grid Architecture Model (SGAM) has been proposed. The main purpose of this architecture model is providing a holistic view on Smart Grid systems [11]. As depicted in Fig. 1, the SGAM is a three-dimensional model and has its origin in the NIST Domain Model [14], the automation pyramid and the GWAC Interoperability Stack [15]. However, due to these specifications, every element within a Smart Grid model can be aligned according to its position in the electricity grid and its role in terms of automation. By doing so, the Domain-axis of the model decomposes a Smart Grid system on basis of the aforementioned NIST Domain Model. On the other hand, the Zone-axis represents the functionality of an element according to automation possibilities with regard to the automation pyramid. To provide interoperability between particular components, five interoperability layers derived from the GWAC Interoperability Stack are introduced. As defined by [11], the layers of the SGAM are explained in the following:

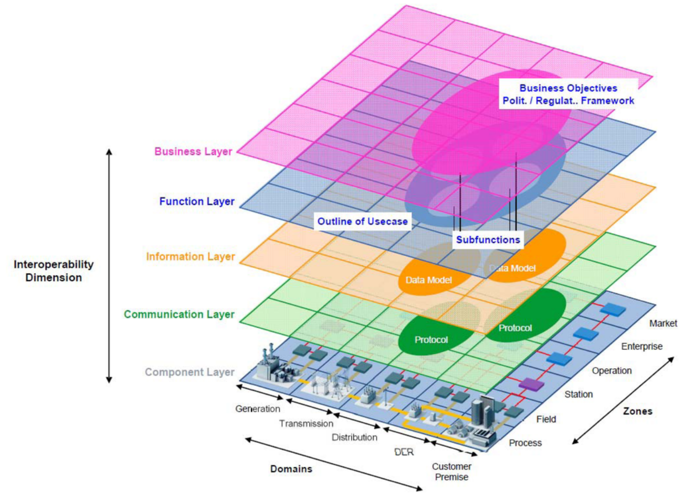

Fig. 1. The Smart Grid Architecture Model (SGAM) [11]

1) Business Layer: Provides a business view on the information exchange related to Smart Grids. Regulatory and economic structures can be mapped on this layer.
2) Function Layer: Describes functions and services including their relationships from an architectural viewpoint.
3) Information Layer: Describes information objects being exchanged and the underlying canonical data models.
4) Communication Layer: Describes protocols and mechanisms for the exchange of information between components.
5) Component Layer: Physical distribution of all participating components including power system and ICT equipment.

From today‘s point of view, the SGAM is widely used as common basis for depicting and architecting Smart Grid systems. Therefore, multiple tools and applications have been developed in order to ensure the applicability of this approach and support the modeling of a Smart Grid system. In [16] the design and implementation of such a tool called the SGAM Toolbox is described, which has established itself as main technology driver to create Smart Grid systems through years of use in international projects by providing domain-specific development features.

B. Domain Specific Systems Engineering

Since the Smart Grid is a widespread and challenging domain, engineering of systems is a complex task and needs to be confronted with suitable methods. According to [17], two disciplines need to be fulfilled. On the one hand, decent knowledge about the domain to operate with should be appropriated, on the other hand, it is mentioned that systems engineering management contributes significantly to the overall success. Concerning these disciplines, an approach called Domain Specific Systems Engineering (DSSE) has been introduced in [18]. It defines an umbrella term for developing systems in particular domains and includes several well-known methods, which are explained in more detail in the following. Thus, to keep the overview of every single aspect included during the engineering of a SoS the concept of Model Based Systems Engineering (MBSE) is usually applied. It enables stakeholders to gain appropriate viewpoints by abstracting the architecture into different levels. Furthermore, it provides technologies to ensure the availability of an iterative development process. The application of the concepts of MBSE must be assured by a suitable modeling language. Due to its freedom, a so-called General Purpose Language (GPL) can be used in a wide variety of application domains. This language with low constraints is tailored to develop systems working in multiple areas. On the other hand, this kind of language is missing specifications for describing detailed processes within a certain area. Therefore, the utilization of a domain specific language (DSL) is unavoidable in order to consider all domain-specific features [19]. To ensure the applicability of MBSE, a well-known approach called Model Driven Architecture (MDA) has been introduced by the Object Management Group (OMG). It makes use of two basic concepts for developing a system, models to selectively present the concerns of a stakeholder and transformations for processing the information to another model [20]. The views specified in MDA are

1) Computation Independent Model (CIM) to provide an understandable description of the system for end users,
2) Platform Independent Model (PIM) to define functionalities and display components of the system,
3) Platform Specific Model (PSM) to formulate interfaces and other technical specifications and
4) Platform Specific Implementation (PSI) to maintain a detailed presentation of code used for describing components within the system.

C. Co-Simulation with Mosaik

Co-simulation is defined as the coordinated execution of two or more models that differ in their representation as well as in their runtime environment [21]. This representation is based on the underlying modeling paradigm where the models may be represented as differential equations while the runtime environment allows those models to be executed. Therefore, individual models can be developed and implemented independently providing an optimal and individual solution. The different simulators that compose the co-simulation are dynamically coupled by using each others input and output variables. Thereby, output variables of one simulator become the input of one or more other simulators and vice versa [22]. The coordination of variable exchanging is facilitated by the master algorithm, which orchestrates the entire co-simulation. A practical example of such a co-simulation framework is Mosaik. Tailored to the Smart Grid, the open source tool is written in Python and integrates a specific power grid simulator like PyPower. By doing so, existing Smart Grid models can be used and instantiated to be processed by simulators with the goal to create large-scale simulation scenarios [10].

III. Approach

As already mentioned, the goal behind this approach is to set up a co-simulation scenario with SGAM and Mosaik in order to investigate the behavior of decentralized IoT-based components and their interplay within the Smart Grid. Since those components inherit considerable complexity themselves, it is crucial to rely on specialists to remain in their field of knowledge in such a SoS. Forcing them to learn a specific modeling language or tool in order to utilize their knowledge tends to end in an overly complex composition of priorities which do not harmonize. However, providing the possibility to remain within their realm of know-how not only raises the accuracy of the individual component but also minimizes the organizational effort necessary to orchestrate the components. Hence, it is a characteristic feature of either SGAM or Mosaik to allow several teams of diverse specializations to work on one large project separately and concurrently. On the other hand, this means, that the scenario itself needs to be developed dynamically considering all influencing factors. Therefore, the method of choice for approaching this is the use of the Agile Design Science Research Methodology (ADSRM). This method emphasizes on the experience gained from the process to amend the next iteration of development and therefore sets a focus on parallel and agile development [23]. The iteration steps of ADSRM adapted to the approach presented in this paper are visualized in Fig. 2. One key advantage of ADSRM is the possibility to enter the development cycle in every of the five phases. In this case, the development is initiated by defining a suitable case study. Within this step, the components as well as the design are defined. Once the case study has been specified the requirements are defined. These requirements are the criterion for the later evaluation step of the approach and therefore represent the experience that can be gained from the process. However, the process for the application itself is modeled to ensure a controlled and structured manner of development. The next stage of ADSRM is the application of the approach onto the case study by applying the process model to its implementation. Finally, the evaluation is done in which the adherence of the previously defined requirements is assessed in order to adjust the case study accordingly for the next iteration.

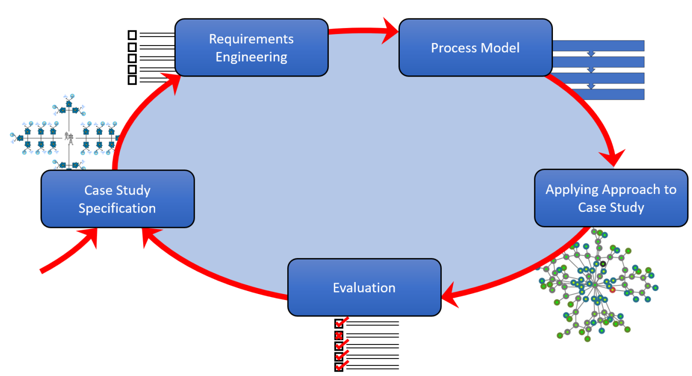

Fig. 2. Agile Design Science Research Methodology (ADSRM) [23]

A. Case Study Design & Requirements

According to ADSRM, the first step is to choose a suitable case study. However, since modeling and simulating an entire energy system would exceed the scope of this work, a suitable extent for an excerpt needs to be derived from a real-world scenario. In this case a typical Smart Grid use case is presented, which is especially focused on charging algorithms of EVs with the intention to reproduce and demonstrate emergent behavior. More precisely, this example makes use of the following components: an electrical power grid including 20 households as well as 20 EVs controlled by a variable charging algorithm. The components are connected with each other through the electrical power grid ensuring a consistent power flow. In addition, all of the 20 households are equipped with several smart home applications utilizing IoT technologies, in order to provide a commensurate load volume and profile.

In the next step, requirements engineering, the design of the co-simulation scenario is elaborated by defining several functional requirements with the intention of evaluating the capability of SGAM to create a model of the aforementioned case study as well as Mosaik to simulate the interplay of its components. Thus, the following requirements have been defined by using methods described in [24]:

1) The system should implement a generic and upgradeable EV simulator. Thereby, the structural and technological composition, specifically the behavior of the battery is of importance. Hence, a comprehensive model of the case study's system architecture including all EVs and smart home components needs to be created at first.
2) The system should be able to visualize and provide the possibility to analyze emergent behavior. More precisely, as described earlier, it should be possible to verify known effects emerging from single components as well as regarding the whole system.
3) The surrounding environment of the system should emulate a modern Smart Grid as detailed as possible. Hence, it is necessary to integrate components like a power flow solver, photovoltaic cells and an electrical power price generator into the simulated model.
4) The system should deal as basis for future developments. The Model and its functional description, like the algorithm to control the charging process of EVs, needs to be adaptable to respond to changes in research and development.

B. Process Model & Co-Simulation Setup

As established by the approach method, the next step is the development of a process model as base for the implementation. Thus, the development process makes use of the requirements in order to manage the process of setting up the co-simulation. By doing so, it is divided into four different phases. First, the required simulation tools are identified. In this case, the choice of tool is the framework Mosaik implementing PyPower as power flow solver and a HDF5 data format. The visualized toolchain is depicted in Fig. 3.

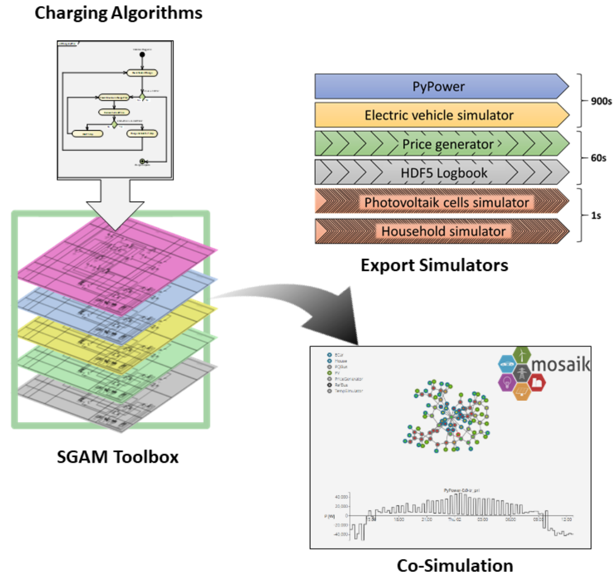

Fig. 3. Toolchain: Architectural Model, Process Model and Mosaik

Subsequently, those tools need to be configured in order to match the defined requirements and fulfil additional ones like tracking the progress, facilitating the data exchange as well as coordinating the respective models, as described in [25]. This phase is followed by the adjustment of the simulators used by the co-simulation in order to operate properly. The main goal is to define a schedule that synchronizes the corresponding step sizes to control the data exchange between each simulator. For households and photovoltaic cells, the step size is defined to be one second due to the changing rate compared to the one of an EV, whose step size is set to be 15 minutes. This value is derived from the entso-e displaying the Austrian load. However, the price generation and the evaluation of data in the HDF5 Logbook are conducted every minute. Finally, the last step of the process model is to integrate all used simulators within an orchestrator, which collects the information and administrates it by its in-built controller.

IV. Application of the Co-Simulation

The first step to realizing the co-simulation is the previous modeling of the case study as a Smart Grid with the help of the SGAM Toolbox. As mentioned earlier, this tool enables the creation of a power system model on basis of the Unified Modeling Language (UML) using the methods of the MDA approach. This means, the whole system is divided into coherent parts as required from the design principles *divide and conquer* and *separation of concerns*. For example, the actual composition of an element is modeled in the Component Layer, the interconnection between those belongs to the Communication Layer. According to this principle, the functionality, more specific the charging behavior of EVs, is located in the Function Layer and represented by activity diagrams, as depicted in Fig. 4. It representatively visualizes the charging cycle of an EV with its dependency to a price.

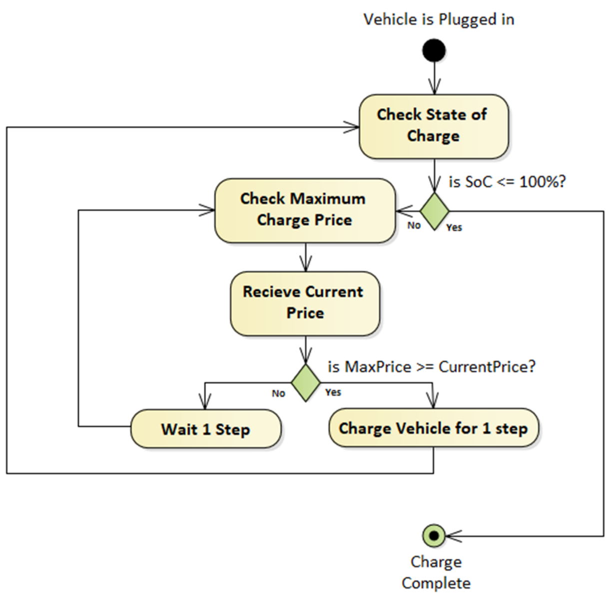

Fig. 4. Activity diagram of EV charging behavior

The depicted process is triggered by the "plug in"-event of the EV. Hence, the goal of the first action is to check the charging state of the battery for querying whether it is at 100% or not. If the battery does turn out to be fully charged, the charge cycle is completed. If the battery however is anything below 100% it is forwarded to the next step, more precisely determining the maximum price at which the EV is allowed to charge. Subsequently the current price for charging is requested so that it can be compared to the EVs maximum price. Therefore, in the following query, it is checked to see whether the current price is still within the price range of the EV. If the price is too high, the process will have to wait for one step and is then diverted back to where the prices are monitored. If the price is low enough for the EV, it will charge for one step before being relayed back to the beginning of the process in which the state of charge is checked. This charging strategy needs to be exported from the model and integrated into Mosaik as an EV simulator. A specifically developed functionality of the SGAM Toolbox is tailored for dealing with this. It provides the possibility to generate source code classes out of the components functional descriptions. Therefore, these classes implement the functionality of the simulator, in this case the charging behavior of the EV. Additionally, the simulator is extended by adding additional information like the actual behavior of a realistic battery, represented in (1).

$$f(x) = \frac{3.7}{\left(\frac{1+x}{95}\right)^{200}} \tag{1}$$

This dynamically adaptable function ensures a static charging process by applying a maximum load of 3.7 kWh, which is used as long as the battery charge reaches 95%. Subsequently, this load decreases steeply until the battery is fully charged. To furthermore implement the price dependency, each model is equipped with a price cap. This cap is generated randomly to ensure a variance and behavioral differences within the models. The price of the price generator is then compared to determine whether the EV model can charge or not. Finally, to create a more realistic "plug in"-behavior, the EVs charging cycle is initially started at random times. As an external component to manage the price, another simulator is then connected to the appropriate counterparts, which reads the current load of the power grid and calculates a price with each step. To ensure that the price does not drop under the theoretical cost of production, another cap was implemented. After implementing the simulators for the households and the photovoltaic cells, a full Smart Grid scenario can be simulated with the existing setup. However, before doing so, the HDF5 database needs to be appended as well in order to prepare the data for the subsequent web visualization.

The web visualization itself offers an interactive interface for selecting and viewing every individual component. This provides not only an insight to the connections that are present, but also to the individual load profile. In Fig. 5 the total load of the power grid is shown in blue together with four EV loads in orange, red, black and green. The inspected time period encloses 36 hours in total. It can be seen how the red EV first started charging around 2:30 p.m., but is not able to resume charging until 11 p.m. due to its individual price cap. This results in a total charge time of 22 hours. Comparing the orange EV to it, which was plugged in at the time the photovoltaic cells where producing power allowing it to charge without a break from 10 a.m. until noon, what results in a total charge time of just about five hours. As time commences, the black and green EVs as well as others not displayed are plugged in, leading to an onwards rising demand for power. The increasing number of plugged-in vehicles therefore leads to an upwards oscillating power grid. Over the course of the night, the vehicles slowly fill up their batteries, which results in a slow decrease of power demand approaching zero just as the households begin to power up and increase their load. Taking this into consideration, the dedicated scenario therefore exemplary indicates the issue of emergent behavior in future energy systems.

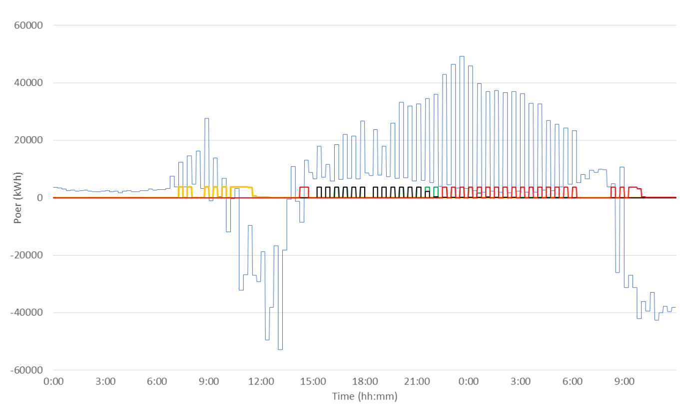

Fig. 5. Example run comparing EVs to total load

V. Conclusion & Future Work

This work introduces a way for modeling and simulating a contemporary Smart Grid including IoT applications with special focus on EVs and the goal to observe emergent behavior. Hence, in the first step a suitable case study is modeled by utilizing the SGAM Toolbox in order to show its context as well as the relation and functionality of its components. Furthermore, an example for simulating this kind of energy system is the tool Mosaik. It is tailored to the particularities of modern power systems and widely applicable with the possibility to display system behavior. However, modeling the characteristics of the system's individual components and applying them in the simulation has proved to be challenging entailing various tasks to accomplish. Thus, the presented approach is especially focused on simulating the charging behavior of EVs and their impact on the Smart Grid in order to not exceed the scope of this work. Nevertheless, as seen in section IV, some adjustments in the architectural model and adding additional functionalities to the SGAM Toolbox made it possible to demonstrate emergent behavior originating from the interplay of these power system components.

The outcome of this work paves the way to stimulate a number of future projects with similar objectives but also alternate intentions with the same underlying concept. For example, this paper does not propose any demand side management technique that would be the method of choice for counteracting with the shown emergent behavior. Applying those more theoretical approaches to a real-world scenario could lead the path towards improving the current state of research. In the same step, a more sophisticated case study could enhance those results by modeling and simulating a considerably more comprehensive Smart Grid. Integrating additional system components and using a substantially larger amount of EVs could result in exposing other unpredictable behaviors as well as the possibility to demonstrate emergent behavior in a large-scale area. Therefore, in order to consider all particularities of such a SoS, the interconnection with models of Smart City or Automotive architectures is essential when developing an approach like this crossing multiple domains.

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] P. Balakumar and S. Sathiya, "Demand side management in smart grid using load shifting technique," in *2017 IEEE International Conference on Electrical, Instrumentation and Communication Engineering (ICEICE)*. IEEE, 2017, pp. 1–6.

[2] H. Andruleit, H. Babies, S. Fleig, S. Ladage, J. Messner, M. Pein, D. Rebscher, M. Schauer, S. Schmidt, and G. Goerne, "Energy study 2016. reserves, resources and availability of energy resources 2011," *Energy Study, Federal Institute for Geosciences and Natural Resources (BGR), Hannover*, 2016.

[3] M. Briggs and K. Sundaram, "Environmentally sustainable innovation in automotive manufacturing and urban mobility (white paper)," *Frost & Sullivan*, 2016.

[4] 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.

[5] X. Liu, B. Gao, C. Wu, and Y. Tang, "Demand-side management with household plug-in electric vehicles: A bayesian game-theoretic approach," *IEEE Systems Journal*, vol. 12, no. 3, pp. 2894–2904, 2018.

[6] P. A. Pattanaik, N. Sahoo, and S. Mishra, "Implementation of demand side management using microcontroller and wireless communication," in *Electrical, Computer and Communication Technologies (ICECCT), 2017 Second International Conference on*. IEEE, 2017, pp. 1–6.

[7] R. Haberfellner, O. de Weck,, E. Fricke, and S. Vössner, *Systems Engineering - Grundlagen und Anwendung*, 13th ed. Orell Füssli, 2015.

[8] D. DeLaurentis, "Understanding transportation as a system-of-systems design problem," in *43rd AIAA Aerospace Sciences Meeting and Exhibit*, 2005, p. 123.

[9] C. Neureiter, S. Rohjans, D. Engel, C. Dänekas, and M. Uslar, "Addressing the complexity of distributed smart city systems by utilization of model driven engineering concepts," in *Proceedings VDE Kongress 2014*, 2014, pp. 1–6.

[10] 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.

[11] S. G. C. CEN-CENELEC-ETSI, "Group.(2012)," *Smart Grid Reference Architecture*, pp. 1–107, 2012.

[12] C. Dänekas, C. Neureiter, S. Rohjans, M. Uslar, and D. Engel, "Towards a model-driven-architecture process for smart grid projects," in *Digital Enterprise Design & Management*, ser. Advances in Intelligent Systems and Computing, P. Benghozi, D. Krob, A. Lonjon, and H. Panetto, Eds. Springer International Publishing, 2014, vol. 261, pp. 47–58.

[13] 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.

[14] 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.

[15] T. Council, "Gridwise interoperability context-setting framework," *GridWise Architecture Council and Battelle Memorial Institute*, 2008.

[16] C. Neureiter, "Introduction to the SGAM Toolbox," Josef Ressel Center for User-Centric Smart Grid Privacy, Security and Control, Salzburg University of Applied Sciences, Tech. Rep., 2013.

[17] B. Lightsey, "Systems engineering fundamentals," DTIC Document, Tech. Rep., 2001.

[18] C. Neureiter, *A Domain-Specific, Model Driven Engineering Approach for Systems Engineering in the Smart Grid*. MBSE4U - Tim Weilkiens, 2017.

[19] A. Van Deursen, P. Klint, J. Visser *et al.*, "Domain-specific languages: An annotated bibliography." *Sigplan Notices*, vol. 35, no. 6, pp. 26–36, 2000.

[20] M. Kempa and Z. A. Mann, "Model Driven Architecture," *Informatik-Spektrum*, vol. 28, no. 4, pp. 298–302, 2005.

[21] 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.

[22] 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.

[23] K. Conboy, R. Gleasure, and E. Cullina, "Agile design science research," in *International Conference on Design Science Research in Information Systems*. Springer, 2015, pp. 168–180.

[24] 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.

[25] A. M. Kosek, O. Lünsdorf, S. Scherfke, O. Gehrke, and S. Rohjans, "Evaluation of smart grid control strategies in co-simulationintegration of ipsys and mosaik," in *2014 power systems computation conference*. IEEE, 2014, pp. 1–7.