Assessing the Suitability of Architectural Models for Generating Smart Grid Co-Simulations

Markus Peter[a], Dominik Vereno[b], Jounes-Alexander Gross[c] and Christian Neureiter[d]
Josef Ressel Centre for Dependable System-of-Systems Engineering,
Salzburg University of Applied Sciences, Urstein Süd 1, 5412 Puch/Salzurg, Austria
{firstname.lastname}@fh-salzburg.ac.at

Keywords: Model Based Systems Engineering, SGAM, Cyber-Physical Energy System.

Abstract: Ensuring the reliability of critical infrastructure, such as a smart grid, is of utmost importance. The verification of this reliability needs to occur early in the systems engineering process. An effective method to accomplish this verification is to simulate a model of a smart grid. Given the complexity of such a system with diverse subsystems, co-simulation has emerged as a leading approach due to its capability to engage various independently developed simulators. This paper explores the interoperability between architectural models and co-simulation. The evaluation relies on a case study implemented both as a simulation and an architectural model, with the goal of identifying similarities and differences. The conclusion drawn is that the two tools do not achieve full interoperability to generate a comprehensive simulation out of an architectural model. This limitation stems from co-simulations requiring precise information at an entity level, which type-based architectural models cannot provide. However, a proposal is put forth to use architectural models as a starting point for generating co-simulation code skeletons. The research provides an analysis of the interoperability challenges and suggests a practical combination of the two concepts.

1 INTRODUCTION

Current power grids face significant problems and limitations, prompting the development of intelligent grids known as smart grids. These smart grids offer a transformative approach to power distribution, responding to the complexities posed by factors such as decentralized energy generation and the increasing diversity of energy sources (Strasser et al., 2020).

The intricate nature of smart grids necessitates system-level validation throughout the entire engineering process (Steinbrink et al., 2017). By simulating different generation and demand patterns, incorporating renewable energy sources, and exploring grid expansion scenarios, decision-makers can assess the feasibility, cost-effectiveness, and reliability of various options. These simulations play a crucial role in optimizing the design and dimensioning of smart grid components, leading to more efficient and resilient systems. Simulation serves as a powerful tool to unravel the complex behaviors and interactions within these systems, offering valuable insights to stakeholders, researchers, and engineers. However, a smart grid, functioning as a system of systems (Pérez et al., 2013), is composed of diverse subsystems. Unlike traditional systems, the components of a smart grid are not centrally developed; instead, they evolve in a decentralized manner with contributions from various entities. This decentralized development introduces complexities when attempting to construct a comprehensive monolithic simulation, as it requires collaboration among different organisations to create a unified model. Co-simulation addresses this challenge by integrating simulation models of different subsystems, allowing them to be described and solved within their respective native environments (Palensky et al., 2017). Importantly, this approach allows organisations to provide their simulators independently, eliminating the need for extensive collaboration.

Despite the advantages of co-simulation, a notable challenge arises in maintaining an overview of the interconnections between different simulators. The intricate nature of the smart grid and the diversity of its components make it challenging to track how various simulators are interconnected.

In response to this challenge, architectural mod-

[a] https://orcid.org/0009-0002-8530-5630
[b] https://orcid.org/0000-0002-7930-6744
[c] https://orcid.org/0000-0002-0351-2111
[d] https://orcid.org/0000-0001-7509-7597

Peter, M., Vereno, D., Gross, J. and Neureiter, C.
Assessing the Suitability of Architectural Models for Generating Smart Grid Co-Simulations.
DOI: 10.5220/0012739800003714

In *Proceedings of the 13th International Conference on Smart Cities and Green ICT Systems (SMARTGREENS 2024)*, pages 38-45
ISBN: 978-989-758-702-3; ISSN: 2184-4968

els could emerge as a bridge, providing a comprehensive and holistic view of the smart grid ecosystem. By abstracting the complexities into well-defined views, architectural models offer an avenue to encapsulate the diverse perspectives and concerns of stakeholders. This approach has the potential to unify the heterogeneous simulation models provided by various stakeholders, enhancing the overall understanding of the system's behavior.

The potential of this approach extends even further when aligned with the European Guide for Power System Testing (Strasser et al., 2020). This guide systematically divides the engineering and validation process of cyber-physical systems like the smart grid into four distinct yet interconnected phases. In the *design phase* typically system-level requirements, application use cases and high-level architecture specification is defined. The *implementation phase* translates these abstract concepts into tangible prototypes, like hardware-in-the-loop configurations or software simulations. In the *validation phase* the developed prototypes are subject to diverse testing. And lastly the *deployment phase marks* the realization of the final product or application, encompassing its integration into the operational environment and subsequent rollout.

Architectural models, with their capacity to provide a comprehensive and abstracted view of the smart grid system, are well-suited for the Design phase. In contrast, the implementation and validation phase can be effectively executed through co-simulation, harnessing the collective capabilities of diverse simulators to conduct thorough and comprehensive testing. It is also worth noting that the authors underscore the current absence of an integrated approach for engineering and validating smart grids (Strasser et al., 2020, p. 4). Therefore, this research aims to bridge this gap by investigating the feasibility of utilizing architectural models as a link to co-simulation within the smart grid domain, contributing to the field by exploring the extent to which architectural models can support the generation of co-simulations in the smart grid context.

This approach not only improves simulation capabilities within the smart grid domain but also holds potential for broader applications in the development and validation of cyber-physical systems across various domains. The insights gained could potentially be adapted and utilized across different domains, thereby expanding the research's impact beyond its initial focus.

2 BACKGROUND

This chapter aims to provide a brief introduction to the concepts of architectural models and co-simulation to establish a shared understanding of these crucial concepts for the research. It aims to ensure a common understanding of terminologies.

2.1 Architectural Models

Conceptual integrity is the most important consideration in system design (Brooks, 1974). One approach to maintaining this integrity is by utilizing architectural models with their holistic perspective and the usage of various viewpoints on the system. Architectural models commonly adhere to the ISO4210 concept (ISO, 2022), outlining requirements for system, software, and enterprise architecture descriptions. The standard defines how system concerns are addressed by different viewpoints independently of technical concepts, modeling languages or tools. In the context of the smart grid domain, the Smart Grid Architecture Model (SGAM) (Smart Grid Coordination Group, 2012) is frequently employed in systems engineering such as in (Uslar et al., 2019) or (Neureiter et al., 2016a). Established through the European Commission's Standardization Mandate it provides a holistic perspective on overall architecture. This model encapsulates methodologies and viewpoints concerning smart grid development, offering a standardized breakdown of a smart grid system with a specific emphasis on interoperability (Bruinenberg et al., 2012). Initially designed to pinpoint gaps in smart grid standardization (Uslar et al., 2019), SGAM has found utility in the broader context of systems engineering.

2.2 Co-Simulation

A general definition of simulation is provided by Loper, who characterizes it as "the execution of a model over time"(Loper, 2015). However, according to Gomez et al., a model is not inherently readily executable; rather, it requires a runtime environment and specific input trajectories (Gomes et al., 2017).

Co-simulation, as defined by (Steinbrink et al., 2017), involves the coordinated execution of two or more models that differ not only in their representation but also in their runtime environment. In this context, a runtime environment is a software system facilitating model execution or solving model equations. For this paper a single pairing of a model with its runtime environment is termed a simulation unit.

The essence of co-simulation lies in multiple simulation units coupled together by a software interface (Vogt et al., 2018). This characteristic enables co-simulation to address various aspects of complex systems, making it particularly well-suited for simulating system-of-systems scenarios, such as those found in the smart grid (Vereno et al., 2023).

2.2.1 Co-Simulation Frameworks

The main components required for a co-simulation framework are the *Scenario*, the *Orchestrator*, the *Simulator* and the *Model Instance* (Barbierato et al., 2022). In Figure 1 the structure of a general co-simulation framework and the co-simulation framework mosaik (Steinbrink et al., 2019a) which will be used in this paper, can be seen.

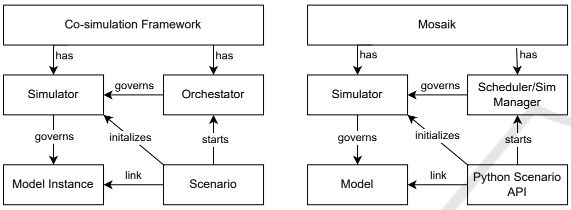

Figure 1: Co-simulation frameworks structure based on (Barbierato et al., 2022).

With Model Instances beeing representations of multiple homogeneous physicial entities, encompassing a physical model that could belong to different mathematical types, finite element methods or behavioural models. Simulators act as Solver, containing a specific Model Instance class. They take on the role of communication adapters with the Orchestrator by instantiating their Models multiple times and overseeing the resulting collection. In practical terms, Simulators transmit inputs received from their counterparts via the Orchestrator and execute Orchestrator commands on their Model Instance collection. In reciprocation, Simulators receive outputs from the Model Instances, sending them back to the Orchestrator for further coordination. The Orchestrator oversees the exchange of data among the Simulators and manages their time regulation and synchronization. And lastly the Scenario serves as a representation of the simulated environment and encapsulates the formal knowledge of the entire cyber-physical system. It embodies the configuration provided by the co-simulation framework, that manages the startup of the Orchestrator, the initialization of the Simulators, and specifying the relationships among Model Instances (Barbierato et al., 2022).

2.3 Current State of Architecture-Based Co-Simulation Generation

The idea of creating simulations based on architectural models is not a new concept. In the automotive field, Binder et al. proposed an approach to explore Industry 4.0 scenarios by implementing them as architectural models and simulating them (Binder et al., 2021). Similar concepts can be found in the smart grid domain, where efforts were made to generate a certain part of a co-simulation out of an architectural model (Binder et al., 2020).

Another study (Binder et al., 2019) focused on developing a prototype interface within a modeling tool to customize simulation settings. However, achieving this required an expansion of the domain-specific language (DSL) used for modeling the architecture. The interface additionally facilitated the mapping of specific activity diagrams to co-simulation models.

Despite the value in exploring this topic, the current state of the art reveals that none of the papers demonstrated complete interoperability between co-simulations and architectural models. The second approach even necessitated adjustments to the DSL itself, underscoring the need for a comprehensive understanding of the interoperability between architectural models and co-simulation. Which further emphasizes the importance of conducting a comprehensive evaluation of the interoperability between architectural models and co-simulation.

3 APPROACH

To adress this research question, general insights can be gained through the developement of a practical artifact connecting these concepts. For this reason the Design-Science Research Methodology (DSRM) (Peffers et al., 2007) was used, since it places strong emphasis on developing a practical artifact as a central component of the research process. DSRM is characterized by iteration cycles, where each iteration contributes to the refinement of our practical artifact. With this paper being at the end of one iteration. Each iteration cycle unfolds through six iterative steps, (Vom Brocke et al., 2020) which can be mapped to different paragraphs of this paper:

- **Problem Identification and Motivation:** In this phase, the research problem is identified, and the value of a solution is justified, corresponding to the introduction of this paper.

- **Defintion of the Objectives for a Solution:** The objectives for a solution will be a concept to automatically generate co-simulations out of architectural models.

- **Design and Developement:** In this phase the key artifact of the study is developed, which will be documented in section 4 and 5.

- **Demonstration:** This activity intends to demonstrate the use of the artifact to solve one or more instances of the project. In this case it can be a prototype which generates a co-simulation or parts of it from architectural models.

- **Evaluation:** In this phase, the effectiveness of the artifact in solving the problem is assessed, as outlined in Section 5.

- **Communication:** Finally, during this phase, all aspects of the problem and the designed artifact are communicated to the relevant stakeholders through the content of this paper.

Now for the developement of such an artifact it is necessary to know which information a co-simulations requires and what information architectural models can provide. To gather this information a case study of a simple smart grid can be conducted and realized both as an architectural model as well as a co-simulation. By comparing the requirements and considerations for both approaches, valuable insights can be obtained concerning the generation of simulations from architectural models.

To develop a simple case study of a smart grid, it is essential to establish a clear definition of what constitutes a smart grid. While different definitions exist, this work follows the definitions provided by (Falk and Fries, 2012) and (SmartGrids, 2006) essentially defining a smart grid as an energy network combined with the bidirectional delivery of energy information to enable a more controllable energy delivery and transmission. Concluding this definition a most basic case study can be divided into the components of an electric grid one or more components using that grid and an component responsible for managing a bidirectional flow of data. One such case study could be the charging behaviour of an electric vehicle in combination with the energy generation of a wind turbine Where a Smart Meter connects these components in a partly bidirectional manner. The scenario entails the charging station purchasing energy from the wind turbine with lower cost than it obtains from the energy market. The basic structure of such can be seen in the Figure 2.

To realize the case study and develop the artifact, two key tools will be utilized. First, for the creation of

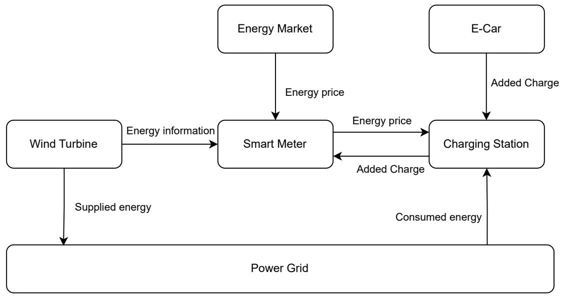

Figure 2: Co-simulaton information flow.

architectural models, the SGAM Toolbox (Neureiter et al., 2016b) will be employed. This modeling tool is grounded in the concept of Model-Based Systems Engineering and closely aligns with the Smart Grid Architecture Model (SGAM), a standardized architecture model for the smart grid domain. The SGAM Toolbox offers a robust foundation for creating architectural models tailored to the specific needs of this research. Secondly, for co-simulation, the study has selected the mosaik co-simulation framework (Steinbrink et al., 2019b). Mosaik is purpose-built for smart grid research and stands out due to its open-source availability and an accessible application programming interface (API). These attributes make it an ideal choice for integrating diverse simulation models within a unified co-simulation environment.

4 IMPLEMENTATION

In this chapter the realisation of the case study as architectural model in the SGAM Toolbox as well as a mosaik co-simulation will be done.

4.1 Realisation as Architectural Model

The modeling process within the SGAM Toolbox, closely aligned with SGAM, employs various models to establish well-defined views across distinct abstraction levels. These viewpoints are designed to encapsulate concerns of different stakeholders. This concept correlates with the SGAM cube, where individual layers can be seen as fundamental viewpoints addressing aspects such as business, functional, informational, communication, and physical facets. Each viewpoint accommodates a grid structure, allowing systematic element placement. Domains along the x-axis represent the electric conversion chain, while the y-axis presents a hierarchical perspective of information management. This can be seen in Figure 3.

The modeling of each layer is done with a domain-specific language relying on UML profiles imple-

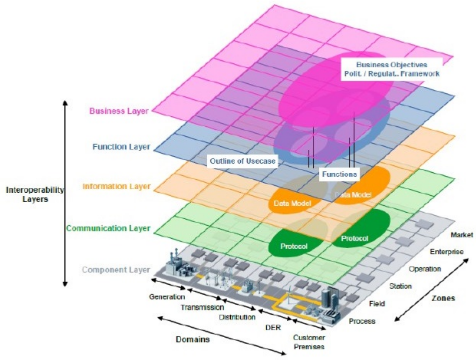

Figure 3: SGAM Framework (Bruinenberg et al., 2012, p. 30).

mented by the toolbox. With UML being a visual modeling language for the architecture, design and implementation of complex software systems. And the DSL being specifically implemented to deliver a basis for domain-related considerations and to create a common basis for all stakeholders (Neureiter, 2017).

The implementation of the layers can be conceptually divided into two main phases: a system analysis phase and a system architecture phase. In the analysis phase, the goal is to provide a comprehensive description of the system's external perspective, encompassing the Business and Function layers. The Business Layer aims to identify all involved business actors along with their specific business goals, while the Function Layer represents functions and the interrelationships between them. The system architecture phase involves the Information-, Communication- and Component layers. The Information Layer is designed to facilitate information exchange, the Communication Layer specifically addresses the definition of communication protocols, and the Component Layer provides a description of the ICT networks employed for communication. In Figure 4, the SGAM Information Layer is depicted, utilizing SGAM-specific components to illustrate the exchange of information among the components in the case study.

4.2 Realisation as Co-Simulation

Implementing a co-simulation within the mosaik framework, as illustrated in Figure 1, involves defining simulation units, each comprising a simulator and a model. Additionally, it is necessary to outline the Scenario, encompassing both the instantiation of simulators and the specification of interconnections among specific model instances. Furthermore the pre-existing simulator PyPower (Scherfke, 2022) is used to implement the structure and characteristics of a power grid. The orchestration of the co-simulation is achieved through the co-simulation framework. In summary, the implementation of the case study, as outlined in Section 3, within the mosaik framework necessitates the development of the following components:

- A scenario file
- A simulator and a model respectively for the following components
  1. Electric Vehicle
  2. Charging station
  3. Smart meter
  4. Wind turbine

The components are developed using the API offered by mosaik. This API is employed by overriding inherited methods from mosaik's simulator, model, and scenario classes. Exemplatory, in listing 1 the overwritten step method for the class ElectricVehicle is illustrated. Within the step method, the process of one model instance's step is defined. In this example of the electric vehicle model, the model determines whether it can accept the provided charge based on its capacity and current charge level.

```
def step(self, loading_factor):
    if self.is_full():
        self.added_charge = 0
    else:
        self.added_charge = self.max_power *
            loading_factor
        self.current_charge = min(self.
            capacity, self.current_charge +
            self.added_charge)

def is_full(self):
    return self.current_charge >= self.
        capacity
```

Listing 1: Step method of the electric vehicle class.

4.3 Simulation Integration

Based on the implementation of the case study a differentiation between fixed syntactic and semantic parts and necessary information to implement different simulations can be made. A listing of the necessary information needed can be seen in table 1.

The information modeled in the SGAM Toolbox can be extracted using the add-in functionality provided by Enterprise Architect (SparxSystems, 2019). Each element modeled in the toolbox is assigned a specific stereotype for identification, allowing its attributes and connections to be saved for further use. The integration of architectural model information into the mosaik simulation follows the approach proposed by Binder et al. (Binder et al., 2019), which

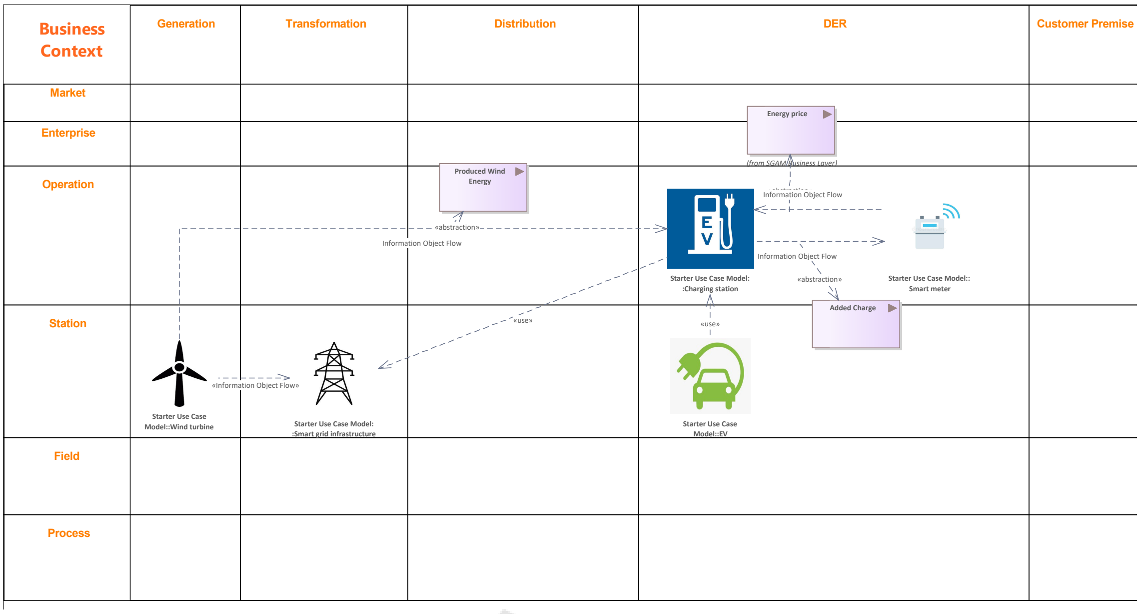

Figure 4: SGAM Information Layer.

Table 1: Necessary information needed for a mosaik co-simulation.

Mosaik component	Necessary information
Simulator	Time behavior Amount of instances Parameter Attributes Power Grid topology Voltage and cable information
Scenario	Simulation runtime Initial values
Model	Model behavior

involves using templates. However, in this iteration, only a code sceleton was generated. This decision was influenced by the limitation of the integrated solution for depicting model behavior, which relied on activity diagrams. These diagrams were considered unsuitable, as explained in more detail in the Evaluation section.

5 EVALUATION

As we proceeded through the necessary steps for generating co-simulations from smart grid architecture models, we encountered two main challenges.

1. The first challenge revolves around the modeling of specific behaviors within the architectural framework. While certain approaches tailored to modeling tools, such as using activity diagrams, may suffice for simpler examples, it's evident that this method isn't universally applicable for all behaviors. Additionally, different simulated entities are often modeled using a variety of tools. For instance, while automotive-specific modeling tools are most effective for modeling charging behavior, they may fall short in adequately capturing wind generation behavior. This needed diversity in modeling tools becomes more complex when considering various stakeholders, each potentially using different tools for their specialized functionality. Addressing this challenge could involve implementing simulators in a standardized format. One such example is the Functional Mockup Interface (Blochwitz et al., 2011), supported by a plethora of tools. This standardization could facilitate seamless integration of diverse simulators, aligning with the overarching goal of uniting independently developed simulators within a single simulation framework.

2. The second challenge lies in reconciling the differing perspectives between architectural models and co-simulations. Architectural models provide a holistic view of a system through type-based models. In contrast, co-simulations necessitate detailed specification of each specific instances of the type based models and their interconnections within given scenarios. However, such specific instance information is frequently absent within architectural models.

While there may be targeted solutions to augment specific tools used for modeling architectural models, such as incorporating an additional instance viewpoint, the ultimate goal is not a one-size-fits-all tool solution. Instead, the emphasis is on leveraging architectural models as a means to integrate independently developed simulators into a broader context. Remarkably, tools like Powerworld (PowerWorld, 2024), employed for power system simulations, possess the capability to export network topology data in auxiliary file formats. Therefore, akin to the aforementioned challenge, a potential solution could involve importing the missing information to bridge the interoperability gap.

Overall after taking a closer look at the interoperability between the architectural models and co-simulation has been taken, the conclusion can be drawn that a full interoperability and thus a complete simulation generation from architectural models as proposed by Binder et al. (Binder et al., 2019) is not deemed to be useful. Architectural models provide a holistic view on the general function of a smart grid in different layers and for different stakeholders. While co-simulations wants to simulate specific use cases of a smart grid and therefore needs specific information about the entity relation of the smart grid components. Therefore instead of forcing an interoperability of concepts with different purposes a combination of both strengths is proposed. Architectural models with their general view on a smart grid can offer a starting point from which simulations of different scenarios could be started. The idea is to generate skeleton co-simulation projects out of architectural models. By utilizing FMU's and integrating them in the models the skeleton code can provide anything but interconnections between the simulation units. The structure of such an approach can be seen in Figure 5.

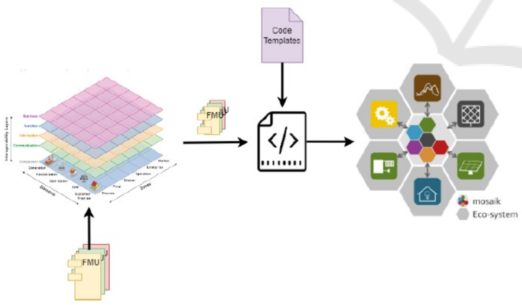

Figure 5: Proposed artifact.

6 CONCLUSION

In conclusion, this paper has looked into the potential of architectural models as a foundation for generating co-simulations within the Smart Grid domain. To assess this interoperability, a case study was designed and implemented as both an architectural model and a co-simulation. Additionally, an artifact connecting these two concepts by generating co-simulation code skeletons from architectural models was developed.

However, this research revealed two notable challenges hindering the seamless generation of co-simulations from architectural models. Firstly, the lack of tool connection, necessitating the specification of various model behaviors in different tools. Secondly, the issue of entity-level modeling, where the necessary instance view for a simulation is not inherently present in the more abstract architectural model.

Despite these challenges, our research underscores the importance of addressing these obstacles to unlock the full potential of architectural models in co-simulation generation. One promising avenue for resolution lies in the utilization of Functional Mockup Units to abstract model behaviors, offering a promising pathway for further exploration and evaluation. Additionally, ongoing efforts, such as the model taxonomy proposed by (Vereno et al., 2024), distinguishing between type and instance-type models, hold promise in tackling the intricacies of entity modeling within architectural frameworks.

In summary, the study highlights the potential of using architectural models for generating co-simulations in the Smart Grid domain. However, it also emphasizes the urgent requirement for ongoing research and innovation to address the challenges we've identified. By overcoming these hurdles, we can fully unlock the power of architectural models as powerful tool for comprehending, analyzing, and enhancing complex systems not only within the Smart Grid but also across various other domains.

REFERENCES

Barbierato, L., Rando Mazzarino, P., Montarolo, M., Macii, A., Patti, E., and Bottaccioli, L. (2022). A comparison study of co-simulation frameworks for multi-energy systems: the scalability problem. *Energy Informatics*, 5(4):1–26.

Binder, C., Agic, A., Neureiter, C., and Lüder, A. (2021). Applying model-based co-simulation on modular production units in complex automation systems. In *2021 IEEE International Symposium on Systems Engineering (ISSE)*, pages 1–6. IEEE.

Binder, C., Fischinger, M., Altenhuber, L., Draxler, D., Lastro, G., and Neureiter, C. (2019). Enabling architecture based co-simulation of complex smart grid applications. *Energy Informatics*, 2(1):1–19.

Binder, C., Fischinger, M., Neureiter, C., Lastro, G., Polanec, K., and Gross, J.-A. (2020). Towards a tool-based approach for dynamically generating co-simulation scenarios based on complex smart grid system architectures. In *2020 IEEE 15th International Conference of System of Systems Engineering (SoSE)*, pages 199–204. IEEE.

Blochwitz, T., Otter, M., Arnold, M., Bausch, C., Clauß, C., Elmqvist, H., Junghanns, A., Mauss, J., Monteiro, M., Neidhold, T., et al. (2011). The functional mockup interface for tool independent exchange of simulation models. In *Proceedings of the 8th international Modelica conference*, pages 105–114. Linköping University Press.

Brooks, F. P. (1974). The mythical man-month. *Datamation*, 20(12):44–52.

Bruinenberg, J., Colton, L., Darmois, E., Dorn, J., Doyle, J., Elloumi, O., Englert, H., Forbes, R., Heiles, J., Hermans, P., et al. (2012). Cen-cenelec-etsi smart grid coordination group smart grid reference architecture. *CEN, CENELEC, ETSI, Tech. Rep*, 23:24.

Falk, R. and Fries, S. (2012). Electric vehicle charging infrastructure security considerations and approaches. *Proc. of INTERNET*, pages 58–64.

Gomes, C., Thule, C., Broman, D., Larsen, P. G., and Vangheluwe, H. (2017). Co-simulation: State of the art. *arXiv preprint arXiv:1702.00686*.

ISO (2022). Software, systems and enterprise — Architecture description. Standard, International Organization for Standardization, Geneva, CH.

Loper, M. L. (2015). *Modeling and simulation in the systems engineering life cycle: core concepts and accompanying lectures*. Springer.

Neureiter, C. (2017). *A domain-specific, model driven engineering approach for systems engineering in the smart grid*. MBSE4U.

Neureiter, C., Engel, D., and Uslar, M. (2016a). Domain specific and model based systems engineering in the smart grid as prerequesite for security by design. *Electronics*, 5(2).

Neureiter, C., Uslar, M., Engel, D., and Lastro, G. (2016b). A standards-based approach for domain specific modelling of smart grid system architectures. In *2016 11th System of Systems Engineering Conference (SoSE)*, pages 1–6. IEEE.

Palensky, P., Van Der Meer, A. A., Lopez, C. D., Joseph, A., and Pan, K. (2017). Cosimulation of intelligent power systems: Fundamentals, software architecture, numerics, and coupling. *IEEE Industrial Electronics Magazine*, 11(1):34–50.

Peffers, K., Tuunanen, T., Rothenberger, M. A., and Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of management information systems*, 24(3):45–77.

Pérez, J., Díaz, J., Garbajosa, J., Yagüe, A., Gonzalez, E., and Lopez-Perea, M. (2013). Large-scale smart grids as system of systems. In *Proceedings of the First International Workshop on Software Engineering for Systems-of-Systems*, pages 38–42.

PowerWorld (2024). Powerworld. https://www.powerworld.com/.

Scherfke (2022). mosaik-pypower: A mosaik extension for simulating power systems with pypower. https://pypi.org/project/mosaik-pypower/. Accessed: 10.12.2023.

Smart Grid Coordination Group (2012). *CEN-CENELEC-ETSI Smart Grid Coordination Group Smart Grid Reference Architecture*.

SmartGrids, E. (2006). Vision and strategy for europe's electricity networks of the future. *European Commission*.

SparxSystems (2019). Enterprise architect 15 im Überblick. https://www.sparxsystems.de/fileadmin/user_upload/pdfs/EAReviewersGuide_EA-15-DE.pdf.

Steinbrink, C., Blank-Babazadeh, M., El-Ama, A., Holly, S., Lüers, B., Nebel-Wenner, M., Ramírez Acosta, R. P., Raub, T., Schwarz, J. S., Stark, S., Nieße, A., and Lehnhoff, S. (2019a). Cpes testing with mosaik: Co-simulation planning, execution and analysis. *Applied Sciences*, 9(5).

Steinbrink, C., Blank-Babazadeh, M., El-Ama, A., Holly, S., Lüers, B., Nebel-Wenner, M., Ramírez Acosta, R. P., Raub, T., Schwarz, J. S., Stark, S., Nieße, A., and Lehnhoff, S. (2019b). Cpes testing with mosaik: Co-simulation planning, execution and analysis. *Applied Sciences*, 9(5).

Steinbrink, C., Lehnhoff, S., Rohjans, S., Strasser, T. I., Widl, E., Moyo, C., Lauss, G., Lehfuss, F., Faschang, M., Palensky, P., van der Meer, A. A., Heussen, K., Gehrke, O., Guillo-Sansano, E., Syed, M. H., Emhemed, A., Brandl, R., Nguyen, V. H., Khavari, A., Tran, Q. T., Kotsampopoulos, P., Hatziargyriou, N., Akroud, N., Rikos, E., and Degefa, M. Z. (2017). Simulation-based validation of smart grids – status quo and future research trends. In Mařík, V., Wahlster, W., Strasser, T., and Kadera, P., editors, *Industrial Applications of Holonic and Multi-Agent Systems*, pages 171–185, Cham. Springer International Publishing.

Strasser, T. I., de Jong, E. C., and Sosnina, M. (2020). *European guide to power system testing: the ERIGrid holistic approach for evaluating complex smart grid configurations*. Springer Nature.

Uslar, M., Rohjans, S., Neureiter, C., Pröstl Andrén, F., Velasquez, J., Steinbrink, C., Efthymiou, V., Migliavacca, G., Horsmanheimo, S., Brunner, H., and Strasser, T. (2019). Applying the smart grid architecture model for designing and validating system-of-systems in the power and energy domain: A european perspective. *Energies*, 12(2):258.

Vereno, D., Harb, J., and Neureiter, C. (2023). Paving the way for reinforcement learning in smart grid co-simulations. In *International Conference on Software Engineering and Formal Methods*, pages 242–257. Springer.

Vereno, D., Polanec, K., Gross, J.-A., Binder, C., and Neureiter, C. (2024). Introducing a three-layer model taxonomy to facilitate system-of-systems co-simulation. In *34th Annual INCOSE International Symposium*.

Vogt, M., Marten, F., and Braun, M. (2018). A survey and statistical analysis of smart grid co-simulations. *Applied Energy*, 222C.

Vom Brocke, J., Hevner, A., and Maedche, A. (2020). Introduction to design science research. *Design science research. Cases*, pages 1–13.