Extending the Concept of Domain Specific Systems Engineering to System-of-Systems

1st Christian Neureiter
Center for Secure Energy Informatics
Salzburg, University of Applied Sciences
Salzburg, Austria
Christian.Neureiter@fh-salzburg.ac.at

2nd Christoph Binder
Center for Secure Energy Informatics
Salzburg, University of Applied Sciences
Salzburg, Austria
Christoph.Binder@fh-salzburg.ac.at

3rd Boris Brankovic
Center for Secure Energy Informatics
Salzburg, University of Applied Sciences
Salzburg, Austria
Boris.Brankovic@fh-salzburg.ac.at

4th Goran Lastro
Center for Secure Energy Informatics
Salzburg, University of Applied Sciences
Salzburg, Austria
Goran.Lastro@fh-salzburg.ac.at

***Abstract*—With the ongoing evolution of cyber-physical systems (CPS) new methods are required to deal with complexity. The concept of Model Based Systems Engineering (MBSE) appears to be a promising approach, even though the common paradigm of object modelling poses a challenge for stakeholders from different domains. To better enable stakeholder integration, in recent past the concept of Domain Specific Systems Engineering (DSSE) has been proposed. This concept aims at establishing a modelling approach around domain specific viewpoints and model kinds. The DSSE approach could have been validated in different domains so far.**

To enable the integration of systems from different domains into System-of-Systems (SoS), this paper contributes conceptual thoughts on how to extend the DSSE approach for application in System-of-Systems Engineering (SoSE). For this purpose, the paper first gives a brief summary on the core concepts of DSSE. Further, a case study is realized to learn about the needs and limitations when it comes to extending DSSE for System-of-Systems Engineering. Finally, identified research items are discussed in more detail and an outlook to our future work is presented.

***Index Terms*—Model Based Systems Engineering (MBSE), System-of-Systems Engineering (SoSE), Domain Specific Systems Engineering (DSSE)**

I. Introduction

Driven by innovation in various fields, development and deployment of *cyber physical systems* CPS as *integrations of computation and physical processes* [1] proceed with fast pace.

Following the definitions given by Haberfellner et al. [2], one could argue that technical systems evolve from *complicated* into *complex* systems. Moreover, individual systems can participate in System-of-Systems (SoS) as characterized for example by Maier and DeLaurentis [3], [4].

Asides the promising capabilities of CPS, some critical aspects need to be taken into account. From a control systems perspective, the ongoing trend towards "fully automated" represents closed-loop scenarios with no human control instance as fall-back scenario anymore. Consequently, such systems are challenged by extensive *dependability* requirements.

Today, when it comes to dealing with dependability in complex systems, the need for modelling is consent within the Systems Engineering (SE) community. The utilization of Model Based Systems Engineering (MBSE) in real-life projects, however, stays behind expectations for several reasons. For example, as criticized by Jean-Marie Favre, conceptual descriptions on the application of models stay rather vague and core concepts for modelling lack a definition through precise models [5]. A different point of view is taken by Edward Lee who argues that *"the role that models play in engineering is different from the role they play in science, and that this difference should direct us to use a different class of models, where simplicity and clarity of semantics dominate over accuracy and detail"* [6].

The contrast of these statements can be resolved by the premise that different models are intended to serve a different purpose. Thus, further differentiation between modelling approaches and models is necessary. Douglas C. Schmidt, for example, differentiates between modelling of "solution space" and "problem space" [7]. On this basis he argues the need for modelling technologies that combine the following:

- *"Domain-specific modelling languages whose type systems formalize the application structure, behaviour, and requirements within particular domains [...]"* [7, p.25]
- *"Transformation engines and generators that analyse certain aspects of models and then synthesize various types of artefacts [...]"* [7, p.25]

The discussion on utilization of models is further enriched by Kent who argues that the aspects of architecture, analysis and process can not be divorced and need to be considered in combination [8]. Kent further identifies the availability of appropriate tooling as crucial prerequisite for efficient engineering and thus, as critical factor.

An attempt to bring the different, mentioned aspects to-

The financial support by the Federal State of Salzburg, Robert Bosch GmbH, Successfactory Consulting Group, and LieberLieber GmbH is gratefully acknowledged.

gether is the concept of Domain Specific Systems Engineering (DSSE). Originating from the Smart Grid application domain, this approach aims at enabling stakeholder participation on the one hand and whilst maintaining a certain degree of rigor on the other hand [9]–[12].

Work on the DSSE approach has started in early 2013 and asides conceptual considerations numerous individual aspects were targeted. Some selected aspects are for example the integration of critical characteristics such as privacy [13]–[15] or security [16]–[18], modelling of business aspects [19], and the development of holistic tool support [13], [20].

From today's perspective the DSSE approach appears to be a step in the right direction. The main artefact originating from this research, the publicly available *SGAM Toolbox*[1] is used in several research and industrial projects. Furthermore, the concept of DSSE could have been successfully transferred to other domains such as *Smart Cities* [21], *Automotive* [22], [23], and Industry 4.0[2] [24]–[26].

One of the cornerstones of the DSSE approach is the integration of a domain's lingua franca within a generic MBSE approach. Main emphasis here is put on domain specific architecture viewpoints and model kinds, accompanied by a generic process model and supported by adequate tool support for modelling or evaluation.

The DSSE approach appears to be fine as long as systems from one particular application domain are considered. The involvement of individual viewpoints and model kinds, however, make it hard when it comes to integration of individual systems into System-of-Systems (SoS) as for example, a Smart Grid architecture model will have other viewpoints than Automotive architectural models. Consequently, this shortcoming in terms of *interoperability* and *compatibility* poses an impediment when it comes for example to investigations on how electric vehicle charging behaviour will affect the overall Smart Grid stability.

To deal with this shortcoming, this paper explores needs and limitations when it comes to extending the DSSE approach to System-of-Systems (SoS). For this purpose, this paper presents a brief summary of the DSSE core concepts in Section II. Subsequently, Section III describes the elaboration of an initial case study that studies the integration of DSSE related models from different domains. The results of this case study are used to identify research items for this attempt which are discussed in Section IV. Finally, Section V summarizes the presented work and outlines our future work.

II. Domain Specific Systems Engineering (DSSE)

To foster the realization of *dependability by design*, the Domain Specific Systems Engineering (DSSE) approach focuses on the establishment of a common system understanding of all involved stakeholders from different disciplines. Thus, a domain-specific Model Based Systems Engineering (MBSE) environment is developed consisting of

[1]www.sgam-toolbox.org
[2]www.rami-toolbox.org

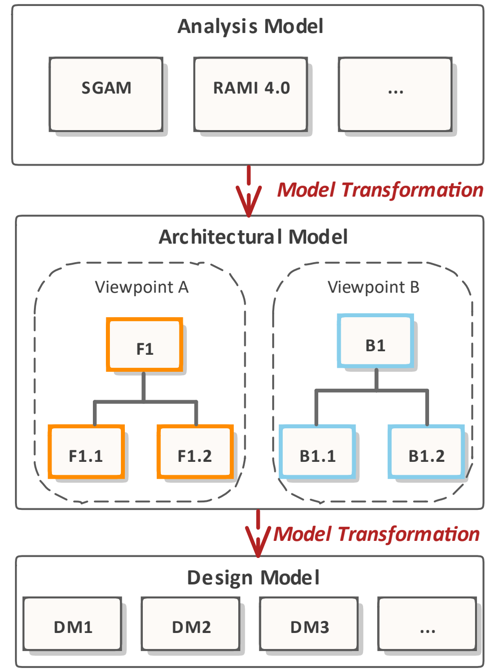

Fig. 1: DSSE modelling approach.

- Domain specific modelling framework
- Process model
- Integrated tool chain

In the following two sections first the modelling framework is described in more detail, followed by a brief summary on the implementation of this approach. As this approach is under development since 2012, different publications exist on various selected topics. A detailed explanation on the DSSE concept as a whole can be found for example in [10], [11], [17].

A. Domain Specific Modelling Framework

As depicted in Figure 1, the DSSE approach provides a modelling framework structured in three stages with explicit model transformations in between.

The first stage (Analysis Model) aims at the establishment of a common system understanding among different stakeholders. For this purpose, well-established concepts from the application domain (e.g., Smart Grids, Industry 4.0, Automotive, and others) are exploited to derive particular *viewpoints* and *model kinds* in accordance to ISO 42010 [27]. To avoid conflicts, at this stage typically reference architecture models from international standardization bodies such as the Smart Grid Architecture Model (SGAM) [28] or Reference Architecture Model Industry 4.0 (RAMI) [29], [30] serve as baseline.

The outcome of modelling at this stage is a clear description of systems involved, their interfaces and associated requirements. Thus, this stage is mainly being used during early development tasks such as requirements development, system context analysis, or elaboration of an initial risk assessment.

After specification and validation of the system and its properties on the domain specific top level, a model transformation takes place. During this step, the individual components of a system are mapped to SysML blocks on the *architectural* level.

The intention of this level is to provide a structured framework for decomposition of the system into its individual design elements. The decomposition is intended to take place on certain well-defined viewpoints. In DSSE as method of choice the SPES methodology [31], [32] has been elected, which covers an iterative decomposition process alongside the four viewpoints *Requirements*, *Function*, *Logical Architecture* and *Technical Architecture*. Similar to the domain-specific stage, individual model kinds are specified for every viewpoint. For example, the function viewpoint has been enriched with the FAS methodology proposed by Weilkiens et al. to guide function development [33].

The last refinement layer of the architectural model yields a description of individual design elements with a clear specification of their requirements, functionality and interfaces. These elements again are transformed into elements on the last stage ("design elements").

On design level, two cases can be separated. First, if the design element is realized by software, a detailed design by means of plain UML can be established. Or, in case of using a specialized technology such as FREDOSAR[3], a horizontal DSL can be utilized to enable the generation of code or modelling artefacts [34].

In the second case, when the design elements are realized by hardware, the detailed specification can serve as input for design engineers from different disciplines. A significant benefit here is the integration of particular design parameters within the particular Design Models (DM) which enables traceability throughout the whole architectural model.

B. Implementation

The implementation of the DSSE concepts comprises a process model, a Domain Specific Language (DSL) and a modelling environment.

For the process model, the concepts of ISO 15288 [35] were aligned with the modelling stages described before. A detailed description of this process model can be found for example in [11].

The modelling environment is implemented as Add-In for the commercial modelling tools *Enterprise Architect*[4] and *Rhapsody*[5]. Backbone of this modelling environment is a DSL, specified by a metamodel. This metamodel comprises an abstract syntax model, a concrete syntax model and a semantic model for every application domain. The implementation of the DSL is done by means of the UML and SysML profile mechanism. Furthermore, the implemented Add-Ins provide a Graphical User Interface (GUI) together with several functionality such as automated model transformations, diagram generation, input- and output functionality, and others.

By now, the modelling environment has been instantiated for four different application domains. In particular, the application domains Smart Grids, Industry 4.0, Automotive Engineering, and Smart Cities are covered. For Smart Grids and Industry 4.0 the implementation could have been done in relation to concepts from standardization. In particular the Smart Grid Architecture Model (SGAM) and the Reference Architecture Model Industry 4.0 (RAMI) were considered. In the field of Automotive Engineering, no common reference architecture could have been found. For this purpose on the one hand the Automotive Reference Architecture Model (ARAM) [22] has been developed and on the other hand, experiments with Software Platform Embedded Systems (SPES) [31], [32] take place. In the field of Smart Cities a reference architecture model is currently being developed by the IEC Systems Committee on Smart Cities [36], [37] and the according toolbox is developed in reference to this work.

Asides our own research, the proposed approach and especially the published toolboxes are being used in context of scientific and industrial projects. It appears as if the concept of "learning the stakeholder's language" instead of forcing stakeholders to learn languages such as SysML could help raising stakeholder acceptance.

However, despite the promising results within individual domains, the DSSE approach falls short when it comes to integration of systems (and models) from different domains. For this reason, concepts are necessary that allow for compatibility and interoperability between different domains as discussed in the following sections.

III. Case Study: Electric Vehicle Integration in the Smart Grid

To better understand challenges and needs for integrating models from different domains, an initial case study has been realized. This case study assumes a scenario where Electric Vehicle (EV) models should be integrated with a Smart Grid model to analyse possible emergent behaviour caused by simultaneous charging. For this purpose, a scenario is assumed where electric vehicles change their charging behaviour in response to a price signal from the grid. The price signal furthermore is calculated on basis of the electric load.

As this case study was intended to investigate mechanisms for System-of-Systems (SoS) construction the models of both, the Smart Grid and the electric vehicles are very limited. Also, interoperability considerations have been limited to the two aspects *functionality* and *electric behaviour*.

In the following, the key aspects of this case study are summarized. A more detailed explanation of this case study can be found in [23], further information on the aspect of Co-Simulation is presented in [20].

As depicted in figure 2, the realized case study comprises the three elements

- Smart Grid Model
- Electric Vehicle Model
- Co-Simulation environment

The model of the Smart Grid has been constructed in reference to the SGAM and by utilization of the SGAM Toolbox. It was limited to only one particular Low-Voltage

[3] www.fredosar.org
[4] www.sparxsystems.com
[5] www.ibm.com

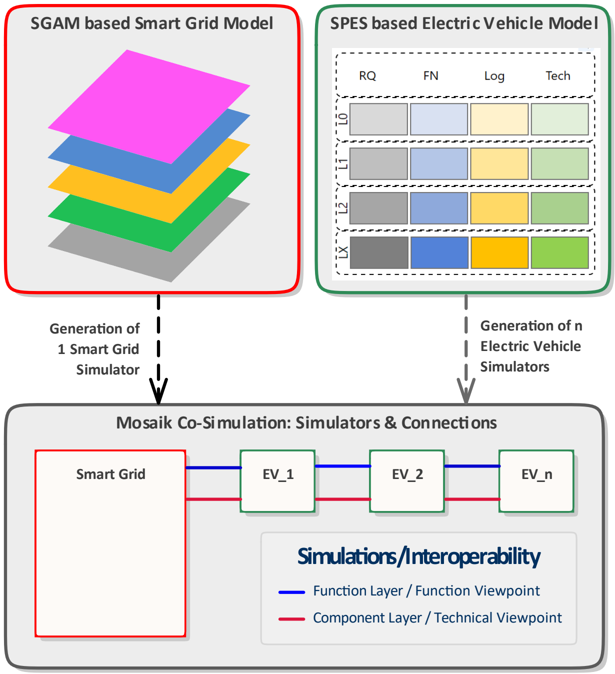

Fig. 2: Setup of the initial case study.

string as part of the Distribution System. Furthermore, functionality has been modelled that changes the tariff in respect to the present electric load and communicates the change to connected electric vehicles. The electric vehicles in that case have been treated as flexible load.

The modelling of electric vehicles has been done on basis of a combination of *ARAM* and *SPES*. Basically, the viewpoints from Software Platform Embedded Systems (SPES) have been combined with model kinds (and concerning model elements) from *ARAM* [22]. The electric vehicles have further been modelled from a functional and a technical perspective. The functional perspective considers both, the communication with the Smart Grid (tariffs) and a simple algorithm that decides if charging should take place. The technical perspective covers a variable electric behaviour in reference to the present charging state.

To enable compatibility and interoperability between these models, different adoptions were necessary. First, from a semantic point of view, the individual interoperability layers of the SGAM have been aligned with the SPES viewpoints as depicted in Figure 3. This simple mapping already reveals certain shortcomings of the SGAM, as for example no explicit integration of requirements or cross-cutting concerns such as security is mentioned. Also, the SGAM Function Layer does not provide clear structure to analyze and develop functions, rather it sticks to Use Cases without further explanation of detailing. For this purpose, as second adoption, both metamodels were extended with the Functional Architecture for Systems (FAS) concept [33] to enable modelling of functionality.

In the technical viewpoint, a model kind “electric model”

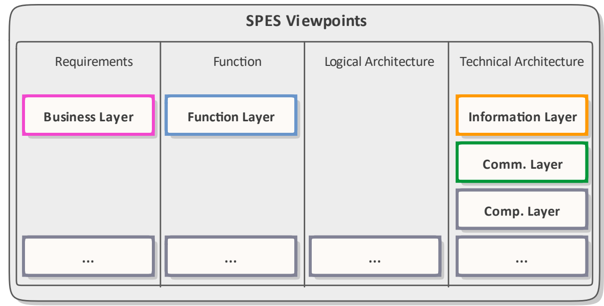

Fig. 3: Alignment of SGAM Interoperability Layers with SPES Viewpoints.

has been integrated to enable an integration of electric behaviour. In this case study, however, this model was limited to the electric power consumption.

All mentioned adoptions have been integrated in the concerning metamodels of the SGAM respectively the SPES toolbox and subsequently, the case study components could have been modelled. These models furthermore could have been used to generate individual simulators that were combined within the MOSAIK Co-Simulation environment. To enable the detection of emergent behaviour, varying number of electric vehicles with a stochastic seed were generated. As mentioned above, more detailed information about the integration of DSSE based models with the MOSAIK Co-Simulation can be found for example in [23] or [20].

The result of this case study was a successful demonstration of emergent behaviour in context of electric vehicle charging. Main focus of this in-depth study, however, was to understand the needs for adoptions of the DSSE approach in order to be used in System-of-Systems (SoS). The following Section summarizes the main findings and discusses particular needs identified during execution of this case study.

IV. Adopting DSSE to System of Systems Engineering

To enable a better understanding and development of System-of-Systems (SoS) architectures, establishment of *interoperability* and *compatibility* between models from different application domains is imperative. For this purpose, the different realizations of DSSE need to be examined to separate *generic* and *domain specific* aspects. As depicted in figure 4, this analysis can be separated into the four dimensions

- Semantics
- Processes
- Methods
- Models and Tools

In the following, a brief discussion on interoperability and compatibility for these dimensions is given.

A. Semantics

During application of the DSSE approach in different domains it came clear that a main barrier for model integration

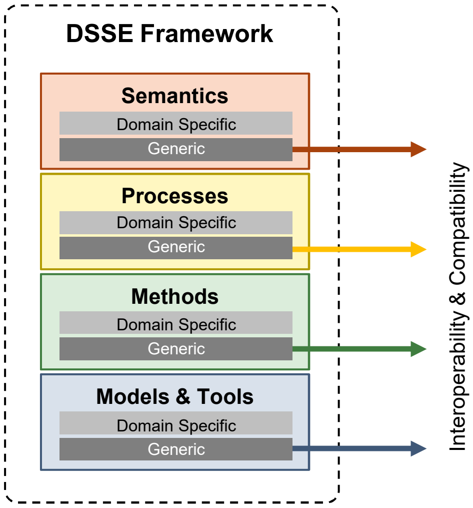

Fig. 4: Generalization of the DSSE Approach.

already can be found in the semantic dimension. Different concepts such as the Smart Grid Architecture Model (SGAM) or Software Platform Embedded Systems (SPES) make use of similar aspects. For example, SGAM knows a *Function Layer* that is intended to represent interaction of individual components or subsystems. A similar concept can be found within the SPES approach as *Function Viewpoint* which hosts a functional decomposition.

Asides the ambiguous definitions and scope of different viewpoints being used, also the terminology of particular elements lacks the necessary rigor. For example, different understanding exists already on basic elements such as *Use Cases*, *Functions* or *Actors*. Especially between hardware and software related disciplines major deviations exist. For example, in mechanical engineering *function development* earns major emphasize and functions are said to carry requirements. In contrast to this, for software related disciplines a function is understood as particular solution for a given requirement.

To enable a seamless integration of models a clear definition of both, a common set of basis viewpoints and a common set of model kinds (with well-defined modelling elements) is required. Furthermore, similar to the ISO/OSI stack for communication in the Internet, an explicit clarification is necessary, which concepts (e.g. use cases, functions, requirements,...) are to be considered in which viewpoint.

B. Processes

Asides the semantic alignment of viewpoints, model kinds and fundamental modelling elements, processes for development require alignment as well. ISO 15288 already delivers a basic set of processes to be addressed in Systems Engineering. However, there is no final answer available so far on how to align development of individual systems and conduct their integration into a superior SoS.

From today's perspective, at least a process for specification of SoS interactions should be mentioned explicitly, complemented with validation aspects. As mentioned before, this aspect remains open and is subject of ongoing research.

C. Methods

Another aspect that could be revealed is the existing divergence in respect to the applied engineering methods. Especially at early stages such as requirements engineering, different approaches are being used. For example, in the Smart Grid community a common approach to start is to collect Use Cases on basis of the IEC 62559 Use Case template as basis for eliciting requirements. In contrast, in automotive engineering the starting point is the collection of different functional and non-functional requirements which, in accordance to SysML concepts, are further refined by utilization of Use Cases. The degree of formalization here also varies and reaches from informal concepts to strongly formalized concepts such as the FAS method [33].

The different approaches being used results in ambiguous understanding of basic modelling concepts. Clearly, there will be no one-fits-all solution but in order to achieve interoperability, at least a taxonomy of the individual elements being used in varying methods would be useful.

D. Models and Tools

The last dimension to be considered is the question of model and tool compatibility. In terms of models, today the application of the DSSE approach to different domains yields similar DSLs. In respect to their genesis in different research projects, however, they are not compatible by nature and individual adoptions are required. To ensure compatibility in the future it will be necessary to develop a common basis that covers the generic concepts on the one hand and delivers the possibility for domain-specific extension on the other hand.

Asides the semantic and technical compatibility between different models, the general question of distributed model development needs to be answered as well. Today, imperfections in standards such as UML or SysML and different interpretations of different tool vendors make it even hard to access standard models with different tools. It is even more challenging, to collaboratively develop a single model. Attempts in this direction exist and progress is being made. However, in the future concepts such as *model slicing* need to gain maturity in order to share models or even pieces of models for enabling System-of-Systems (SoS) considerations.

V. Summary and Future Work

Evolution of System-of-Systems proceeds with fast pace and holistic engineering concepts are required to enable the integration of *dependability* by design. The paper at hands aims at extending the Domain Specific Systems Engineering approach for application in System-of-Systems Engineering. Thus, after an initial discussion on the core concepts of DSSE, basic needs and limitations for this attempt were studied by elaboration of a first case study. This in-depth study reveals an initial set of work items in the four dimensions semantics, processes, methods, and models and tools.

Asides considerations on processes and methods, the initial task for investigation is the definition of a common set of generic viewpoints and model kinds as basis for semantic interoperability. Further, a consolidation of different DSSE related toolboxes is necessary to integrate these aspects and validate the concept. In reference to this prioritization, our present work concentrates on the development of a so called *Meta DSL*. This Meta DSL is intended to enable the development of interoperable modelling environments on basis of common concepts. Having the possibility for integrating models from different domains, explicit emphasis can be put on the development of dedicated concepts that enable a better understanding and architecting of dependable System-of-Systems (SoS) architectures.

References

[1] E. A. Lee, "Cyber physical systems: Design challenges," in *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*, May 2008, pp. 363–369.

[2] R. Haberfellner, O. L. de Weck, E. Fricke, and S. Vössner, *Systems Engineering. Grundlagen und Anwendung*. Orell Füssli, 2012.

[3] M. W. Maier, "Architecting principles for systems-of-systems," *Systems Engineering*, vol. 1, no. 4, pp. 267–284, 1998.

[4] D. DeLaurentis, "Understanding Transportation as a System-of-Systems Design Problem," in *43rd AIAA Aerospace Sciences Meeting and Exhibit. Reno, Nevada*, 2005.

[5] J.-M. Favre, "Towards a Basic Theory to Model Driven Engineering," in *3rd Workshop in Software Model Engineering, WiSME*. Citeseer, 2004, pp. 262–271.

[6] E. A. Lee, "Fundamental limits of cyber-physical systems modeling," *ACM Trans. Cyber-Phys. Syst.*, vol. 1, no. 1, pp. 3:1–3:26, Nov. 2016.

[7] D. C. Schmidt, "Model-Driven Engineering," *IEEE Computer Society*, vol. 39, no. 2, p. 25, 2006.

[8] S. Kent, "Model Driven Engineering," in *Proceedings on the 3rd International Conference on Integrated Formal Methods*. Springer, 2002, pp. 286–298.

[9] 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.

[10] 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 the 11th International Conference on System of Systems Engineering (SoSE)*. IEEE, 2016, Best Paper Award.

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

[12] M. Uslar, S. Rohjans, C. Neureiter, F. Proestl Andren, J. Velasquez, C. Steinbrink, V. Efthymiou, G. Migliavacca, S. Horsmanheimo, H. Brunner, and T. Strasser, "Applying the smart grid architecture model for designing and validating system-of-systems in the power and energy domain: A european perspective," *Energies*, vol. 12, no. 2, 2019.

[13] C. Neureiter, D. Engel, J. Trefke, R. Santodomingo, S. Rohjans, and M. Uslar, "Towards Consistent Smart Grid Architecture Tool Support: From Use Cases to Visualization," in *Proceedings of the 5th International Conference on Innovative Smart Grid Technologies Europe (ISGT Europe)*. IEEE/PES, 2014, pp. 1–6.

[14] F. Knirsch, D. Engel, C. Neureiter, M. Frincu, and V. Prasanna, "Model-driven privacy assessment in the smart grid," Josef Ressel Center for User-Centric Smart Grid Privacy, Security and Control, Technical Report 2014-01, Jul. 2014.

[15] ——, "Privacy Assessment of Data Flow Graphs for an Advanced Recommender System in the Smart Grid," in *Information Systems Security and Privacy – Revised and Selected Papers of ICISSP 2015*, ser. Communications in Computer and Information Science, O. Camp, E. Weippl, C. Bidan, and E. Aïmeur, Eds. Springer International Publishing, 2016, vol. 576, pp. 89–106, Best Paper Award.

[16] C. Neureiter, G. Eibl, D. Engel, S. Schlegel, and M. Uslar, "A concept for engineering smart grid security requirements based on SGAM models," *Computer Science - Research and Development*, pp. 1–7, 2014.

[17] C. Neureiter, D. Engel, and M. Uslar, "Domain specific and model based systems engineering in the smart grid as prerequesite for security by design," *Electronics*, vol. 5, no. 2, p. 24, 2016.

[18] M. Uslar, M. A. van Amelsvoort, C. Delfs, D. Engel, C. Neureiter, and C. Nabe, Tech. Rep.

[19] M. Pavlovic, T. Gawron-Deutsch, C. Neureiter, and K. Diwold, "SGAM business layer for a local flexibility market," in *Proceedings of the CIRED Workshop 2016*. Institution of Engineering and Technology, Jun. 2016, pp. 221–224.

[20] 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, to appear.

[21] 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*, Oct. 2014, pp. 1–6.

[22] D. Draxler, C. Neureiter, G. Lastro, T. Schwartzkopff, and M. Boumans, "A domain specific systems engineering framework for modelling electric vehicle architectures," in *2019 IEEE Conference and Expo Transportation Electrification Asia-Pacific (ITEC Asia-Pacific)*. Jeju, Korea: IEEE, 2019, Best Presentation Award.

[23] 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 *14th Annual Conference System of Systems Engineering (SoSE)*. Anchorage, Alaska, USA: IEEE, 2019.

[24] C. Binder, D. Draxler, C. Neureiter, and G. Lastro, "Towards a Model-Centric Approach for developing Functional Architectures in Industry 4.0 Systems," in *5th IEEE International Symposium on Systems Engineering (ISSE)*, Edinburgh, Scotland, UK, 2019, to appear.

[25] C. Binder, C. Neureiter, and G. Lastro, "Towards a Model-Driven Architecture Process for Developing Industry 4.0 Applications," *International Journal of Modeling and Optimization*, vol. 9, no. 1, pp. 1–6, 2019.

[26] C. Binder, C. Neureiter, G. Lastro, M. Uslar, and P. Lieber, "Towards a standards-based domain specific language for industry 4.0 architectures," in *Complex Systems Design & Management*, E. Bonjour, D. Krob, L. Palladino, and F. Stephan, Eds. Cham: Springer International Publishing, 2019, pp. 44–55.

[27] International Organization for Standardization, *ISO/IEC/IEEE 42010: Systems and software engineering – Architecture description*, International Standards Organization Std., 2011.

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

[29] M. Hankel and B. Rexroth, "The Reference Architectural Model Industrie 4.0 (RAMI 4.0)," *ZVEI*, 2015.

[30] ZVEI - Zentralverband Elektrotechnik- und Elektronikindustrie e.V., "Industrie 4.0: Das Referenzarchitekturmodell Industrie 4.0 (RAMI 4.0)," ZVEI, Tech. Rep., 2015.

[31] K. Pohl, H. Hönninger, R. Achatz, and M. Broy, *Model-Based Engineering of Embedded Systems: The SPES 2020 Methodology*. Springer, 2012.

[32] K. Pohl, M. Broy, H. Daembkes, and H. Hönninger, *Advanced Model-Based Engineering of Embedded Systems: Extension of the SPES 2020 Methodology*. Springer, 2016.

[33] T. Weilkiens and L. Jesko, *Model-Based System Architecture*, Systems Engineering and Management ed. Wiley and Sons, 2015.

[34] M. Fischinger, C. Neureiter, C. Binder, N. Egger, and M. Renoth, "Fredosar: Towards a security-aware open system architecture framework supporting model based systems engineering," in *8th International Conference on Smart Cities and Green ICT Systems (SMARTGREENS)*. Heraklion, Crete - Greece: SciTePress, 2019, pp. 108–115.

[35] International Organization for Standardization, *ISO 15288:2015 Systems engineering - System life cycle processes*, Std., 2015.

[36] IEC, "IEC TS 63188 ED1: Systems Reference Document - Smart Cities - Smart Cities Reference Architecture Methodology (SCRAM)," IEC, Tech. Rep., 2019.

[37] International Electrotechnical Commission, *IEC TS 63188 ED1: Systems Reference Deliverable - Smart Cities - Smart Cities Reference Architecture Methodology (SCRAM)*, Std., 2019.