Review on Domain Specific Systems Engineering

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

***Abstract*—The ongoing integration of connectivity and automation in cyber-physical systems (CPS) drives the need for innovative engineering concepts. Especially the aspect of dependability by design comes into focus. A suitable approach for this task can be found in the concepts of Model Based Systems Engineering (MBSE). In practical application, however, utilization of MBSE doesn't meet expectations. One of the main barriers identified is the little stakeholder acceptance for common concepts such as object modelling. This aspect has been addressed in the recent past by the Domain Specific Systems Engineering (DSSE) approach. In context of DSSE, plenty of research has been conducted and different aspects were studied. To gain a holistic picture of this approach, the paper at hands contributes a review on DSSE as a whole. In a first step, the main concepts of Domain Specific Systems Engineering are summarized and the application of this approach is discussed. Subsequently, research outcomes achieved so far are consolidated and discussed in detail. On this basis, remaining issues are identified and the future agenda for research in DSSE is outlined.**

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

I. INTRODUCTION

The ongoing integration of innovative technologies such as Artificial Intelligence (AI) or 5G drives the evolution of cyber-physical systems (CPS) with fast pace. As outlined by Lee, the term CPS summarizes the *integration of computation and physical processes* [1] which makes the need for certain aspects of dependability inherent.

The integration of dependability, however, is challenged in various ways. Based on the classification proposed by Haberfellner et al. [2] one could argue that CPS evolve towards *complex* systems. Or, even more, by following the considerations from Maier [3] and DeLaurentis [4], such systems can be part of superior System-of-Systems (SoS). Besides increasing complexity, realization of dependability is further challenged by the interdisciplinary character of such systems that requires close cooperation between stakeholders from different disciplines and domains.

A natural concept for dealing with complexity is the utilization of modelling. Generally spoken, modelling addresses complexity by application of the two fundamental paradigms *abstraction* and *separation of concerns*. Thus, for developing systems by means of modelling, a rigorous concept of aligned viewpoints, model kinds, tools, and processes is deemed crucial. The genesis of model-based approaches, however, did not follow a single (research) track. Rather, a plethora of different approaches from different domains and disciplines came up and consequently different terminology, concepts and understanding exists. As pointed out in a recent publication by Hick, this lack of understanding already starts with the definition of fundamental terms such as *system model* [5].

The missing rigor in model-based approaches has been criticised for example by Jean-Marie Favre. Particularly in context of Model-Driven Engineering (MDE) Favre points out: *Though MDE is supposed to be about precise modelling, MDE core concepts are not defined through precise models. One can find plenty of metamodels in the literature to describe particular technologies, or tools, but we are not aware of a single one that fully captures the MDE notions at a global level.* [6].

Even though the existing challenges and imperfections in model-based approaches are widely acknowledged, the future direction remains subject of discussion. Edward Lee for example argues that models in engineering play a different role than in science and consequently, a new class of models should be considered where *simplicity and clarity of semantics dominate over accuracy and detail* [7].

The ongoing discussion on models has further been addressed by Schmidt who argues in favour of a differentiation between problem space and solution space [8]. As Schmidt points out, domain-specific modelling languages that formalize the application domain should be enriched with transformation engines and generators that allow for synthesizing various types of artefacts as part of the solution space.

Today it appears as if the term Model Based Systems Engineering (MBSE) turns out to be the least common denominator. Defined by INCOSE[1], MBSE addresses *the formalized application of modelling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases* [9].

In practical application, the implementation of MBSE is often centred around object models created by utilization of

---

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

[1] www.incose.org

OMG's[2] Systems Modelling Language (SysML) [10]. Despite the vast capabilities of object modelling in general and SysML in particular, this approach requires good understanding of the underlying concepts such as object orientation. This aspect poses a relevant barrier for acceptance among several stakeholders and limits the application of this approach.

To enable a better integration of stakeholders from different domains and disciplines, the concept of Domain Specific Systems Engineering (DSSE) has been proposed. This approach aims at enabling seamless modelling throughout the whole engineering cycle. Further, an accompanying modelling environment has been developed that supports architecture development, evaluation and realization.

Research on DSSE started more than eight years ago and since that, different aspects have been studied in detail. Moreover, the approach has been validated in different application domains such as Smart Grids, Industry 4.0, Automotive Engineering, or Smart Cities.

To enable a general reflection on DSSE, the paper at hands contributes a review of the concept, research conducted so far and a detailed discussion on outcomes and necessary work. Thus, Section II summarizes the fundamental concepts of DSSE before Section III discusses its application in different application domains. Subsequently, Section IV provides a detailed review of the DSSE approach in reference to individual research aspects. Finally, Section V summarizes this paper and drafts our research agenda for the future.

II. DOMAIN SPECIFIC SYSTEMS ENGINEERING (DSSE)

The basis for Domain Specific Systems Engineering (DSSE) has been layed in *Smart Grid* related research where concepts were investigated to enable *dependability by design*. Since that, the approach continuously evolved and the concepts were transferred to complementary application domains.

Basically, DSSE aims at providing a framework for implementation of Model Based Systems Engineering (MBSE). Thus, it comprises the three building blocks *Process Model*, *DSSE Modelling Framework*, and *DSSE Modelling Environment* as depicted in Figure 1. Further, these building blocks are aligned with the four layers of Model-Driven Architecture (MDA) [11], [12] as specified by Object Management Group (OMG).

The ideas behind the individual building blocks and their interrelation are summarized in the following sections. More detailed information on the overall concept can be found for example in [13], [14], or [15].

A. Process Model

As DSSE strives to cover all engineering phases, the existence of a holistic process model is required. For this purpose, the process model from ISO 15288 [16] has been aligned with MDA, yielding the three main phases *System Analysis*, *System Architecture*, and *Design and Development*.

The initial phase (System Analysis) corresponds to MDA's Computation Independent Model (CIM) and aims at establishing a common understanding of the system in its context, the stakeholders involved and the corresponding requirements. Subsequently, the System Architecture phase yields the Platform Independent Model (PIM) focusing on a technology-neutral decomposition of the system. Realization of particular elements takes place during the last stage of this process model, the Design & Development phase. This phase is intended to be executed iteratively and incremental and yields - in an alternating manner - MDA's Platform Specific Model (PSM) and Platform Specific Implementation (PSI).

Every project phase covers processes, activities and outcomes described by ISO 15288. As suggested by ISO, this process model is not intended to serve as one-fits-all solution. Rather, it represents a starting point for individual adoption. To enable such adoptions, this process model has been modelled by utilization of the Software & Systems Process Engineering Meta-Model (SPEM) [17] notation. Further, outcomes from this process model reference particular artefacts from the *DSSE Modelling Framework* which is described in the following section.

An example for adoption of the ISO 15288 process model together with more detailed information on the underlying thought model can be found for example in [14] or [15].

B. Modelling Framework

The DSSE Modelling Framework defines various *viewpoints* and *model kinds* in reference to the concepts of ISO 42010 [18]. In context of DSSE a model kind represents a collection of particular diagram types, model elements and their relations. Further, *functionality* is defined that can be applied to either a single element or portions of the whole model.

The specification of the DSSE Modelling Framework is done on basis of a metamodel consisting of an Abstract Syntax Model (ASM), a Concrete Syntax Model (CSM), and a Semantic Model (SM).

The ASM as backbone of the metamodel defines *viewpoints*, *model kinds*, *modelling elements*, and their corresponding *relations* as type graph, complemented with formal constraints for instantiation. The type graph is further enriched with the specification of *functionality* allocated to particular elements. The resulting type graph serves as input for subsequent development of a Domain Specific Language (DSL) whereas *functionality* contributes to the product backlog for implementation of the *DSSE Toolbox*.

Finally, the ASM is complemented with the CSM and the SM which define the notation (visual appearance) of particular elements respectively the semantics behind.

The organization of the metamodel further reflects the structure of MDA. Thus, elements are associated with particular stages and explicit *model transformations* are specified to enable a mapping between.

On top of the DSSE modelling framework the *Analysis Model* focuses on establishment of common understanding of the system, its context and the required functionality.

---

[2] www.omg.org

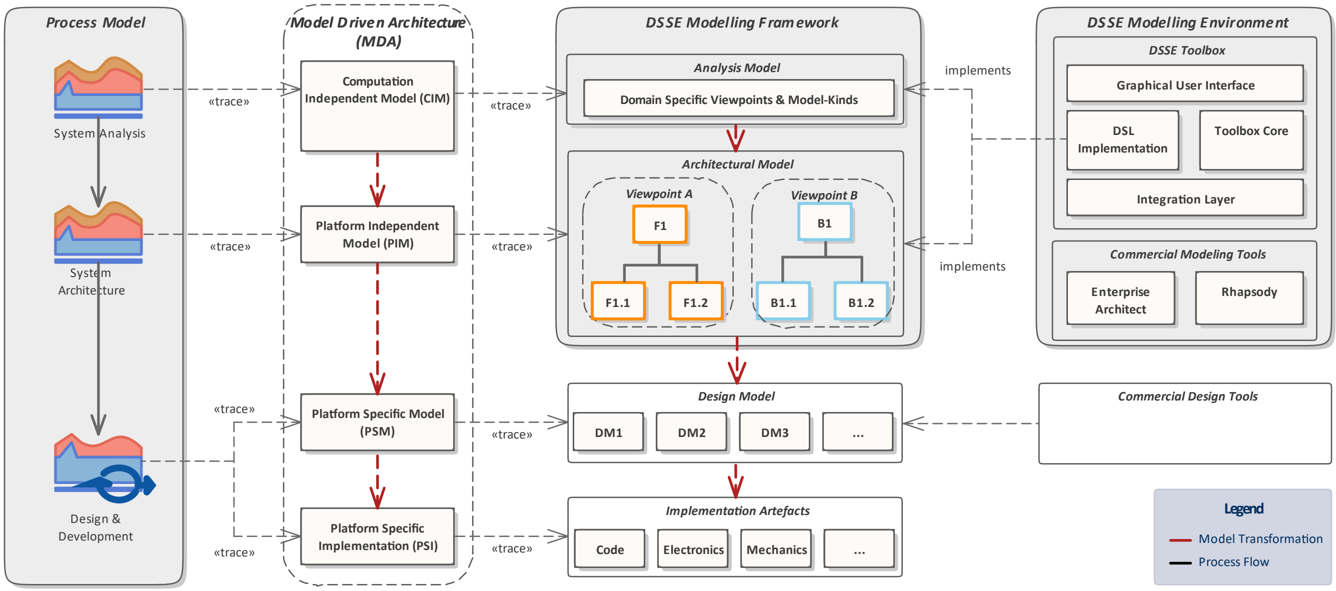

Fig. 1: Overview on the DSSE Approach

Corresponding to MDA's CIM, the system is considered as black box model. Viewpoints and model kinds being used on this stage are closely related to terminology that is well established within the application domain.

In the Smart Grid, for example, viewpoints are derived from the Smart Grid Architecture Model (SGAM) as proposed by standardization bodies [19]. The SGAM as reference architecture model proposes a structured framework for considering business, function, information, communication, and component (physical) aspects. For utilization in engineering, however, the complementation with additional viewpoints such as *requirements* is necessary.

Subsequent to the analysis phase, the *Architectural Model* utilizes the object modelling paradigm to conduct a system decomposition. The Architectural Model corresponds with MDA's PIM and realizes an iterative decomposition of the system. As outcome, particular *design elements* are identified that can be handed over to different design engineers.

The decomposition is conducted in reference to the Software Platform Embedded Systems (SPES) methodology [20], [21]. Therefore it is done alongside the four viewpoints *Requirements*, *Function*, *Logical Architecture*, and *Technical Architecture*. As model kinds, mainly collections of SysML diagrams are being used with an exception in the function viewpoint. Due to the lacking possibility for *function development* within native SysML, the language is extended with the Functional Architecture for Systems (FAS) methodology [22]. This approach describes a step-by-step process to derive particular functionality on basis of Use Cases and to allocate it to functional elements.

As outcome of the decomposition process a set of particular *design elements* is developed, specified as SysML blocks with well-defined interfaces, behaviour, and requirements. Further, the most important design parameters are identified and related to design element requirements. Due to the nature of object modelling, individual design parameters can be traced throughout the whole model which enables investigation of side-effects when changing a particular component.

The *Design Model* covers the detailed design of particular design elements and can be related with MDA's PSM. Depending on the nature of a particular design element, the detailed design can be an electrical schematic, a CAD construction model, a particular piece of software design, and others. A special case, however, is the integration of source code generation. As demonstrated for example in [23], the existence of a suitable code framework such as FREDOSAR[3] enables a seamless integration of both, the detailed design and the implementation artefacts within the DSSE framework.

C. Modelling Tool

To enable the application of the DSSE Modelling Framework, the *DSSE Toolbox* has been implemented as add-in for commercial modelling tools such as Enterprise Architect[4] or Rhapsody[5]. As depicted in Figure 1, the architecture of the toolbox consists of four building blocks.

Access to the model structure of commercial tools is realized by the *Integration Layer* which is implemented individually for every considered tool. Backbone of the toolbox is the logic layer, comprising the two building blocks *DSL Implementation* and *Toolbox Core*. The DSL implementation realizes the metamodel as described earlier by utilization of the UML profile mechanism. Some more effort is put in the Toolbox Core that implements two aspects. First, general

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

functions such as model management, installation routines, or interfacing capabilities and second, functionality described within the metamodel.

Finally, the last building block is a Graphical User Interface (GUI) that guides through the modelling process and provides access to various functionality.

At present, two different implementations of the toolbox exist. One is written in C# as add-in for *Enterprise Architect*, the second is implemented as JAVA based extension for *Rhapsody*. Both implementations of the toolbox have been instantiated for various application domains. Further, the particular implementations for Smart Grids (*SGAM Toolbox*[6]) and Industry 4.0 (*RAMI Toolbox*[7]) are publicly available. Some more detailed information on the toolbox concepts can be found for example in [14], [15], [24], or [25].

III. Application and Research Items

Since the first attempts in context of Smart Grids, the DSSE approach has been transferred to different application domains. Further, several research items have been identified and studied in detail. In the following two sections, first the application of DSSE and second, considerations on particular aspects are summarized.

A. Application of DSSE

Besides customization of the process model, main adoptions of DSSE for a certain application domain take place within the Modelling Environment. In particular, the analytic model needs to be adopted to meet an application domains lingua franca.

Most research work in this context has been done in context of *Smart Grids*. The main benefit in this application domain is the existence of the Smart Grid Architecture Model (SGAM) as standardized reference architecture model [19]. Proposed by standardization bodies, this concept finds broad acceptance among several stakeholders and thus, provides a well-suited basis for common understanding.

The genesis of the DSSE approach within this particular application domain is well documented by several publications such as [13]–[15], [24], [26]–[29], or [30].

The first attempt to transfer the approach to other domains took place in context of Industry 4.0 where a reference architecture model, similar to SGAM, has been proposed. Basically, the Reference Architecture Model Industry 4.0 (RAMI) has been derived from SGAM which supports easy adoption. A detailed discussion on application of DSSE in Industry 4.0 can be found for example in [31]–[33], or [34].

Despite the promising results in the domains Smart Grids and Industry 4.0 it has to be mentioned that the existence of well accepted reference architecture models from standardization bodies cannot be taken for granted. In that case, conceptual work is necessary to identify and/or establish a common language first.

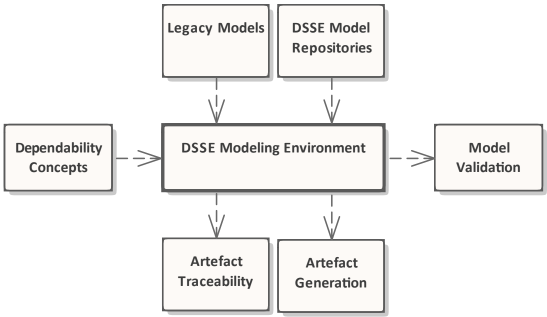

Fig. 2: Envisioned Application Scenario for DSSE

One example here is the rather unspecific field of *Smart Cities*. To address the heterogeneity of this domain a generic concept has been developed, capable of addressing various aspects [35]. In the meanwhile, however, IEC as standardization body has started drafting a *Smart City Reference Architecture (SCRA)* [36], [37] which we will integrate with the DSSE approach in the near future.

Another example considered is the application domain of *Automotive Engineering*. Other than expected, no commonly accepted reference architecture for vehicles could be found in literature. Thus, for application of DSSE in a first step the Automotive Reference Architecture Model (ARAM) has been developed. Even though main parts of this work are subject of a non-disclosure agreement, conceptual information on the establishment and application of ARAM can be found for example in [38] or [39].

B. Selected Research Items

Complementary to the focus on different domains, various technical aspects have been studied in depth. To provide a reference for the subsequent discussion of individual research items, Figure 2 depicts an overview on the envisioned application of the DSSE approach. More detailed information on the underlying concepts and toolchain can be found for example in [14], [15], [25], or [24].

Architecture development barely starts from scratch which draws the need for integration of existing artefacts. As depicted in Figure 2, import of of *Legacy Models* and import from *DSSE Model Repositories* have been studied. In terms of *Legacy Models* the import of existing work created with different tools is targeted. For example, the network topology of a power grid could serve as starting point for architecting Smart Grid functionality. To study this aspect, attempts have been made for interfacing with various tools from the power system domain. The lack of common interface standards, respectively their varying interpretation, however, requires significant customization effort for every individual tool though.

Other than legacy models, the integration of *DSSE Model Repositories* is intended to offer library functionality specific for DSSE. These libraries could cover common concepts such as Use Cases, reference solutions, or products from a vendor's portfolio. One example for such a repository is the *Smart*

---

[6] www.sgam-toolbox.org

[7] www.rami-toolbox.org

Grid Use Case Management Repository (UCMR) as discussed for example in [40], [41] or [42]. Both, the UCMR and the DSSE metamodel follow the specification scheme proposed by IEC 62559 [43] which makes semantic compatibility inherent. More effort, however, is the establishment of content for such libraries which became apparent during our attempts to establish a repository of privacy-relevant Smart Metering Use Cases [44].

A key driver for work on DSSE is to support the realization of dependability *by design*. Therefore, studies took place how to integrate particular aspects of dependability such as *privacy* or *security*. We are aware that privacy typically is not considered a dependability characteristic, as proposed for example by Avizienis et al. [45]. However, due to the uprising importance, we argue to consider privacy with similar seriousness [46]. Due to the novelty of this aspect, mainly conceptual work on integration took place as discussed for example in [25].

Contrasting to privacy, the aspect of *security* can rely on mature work such as the *NIST Guidelines for Smart Grid Cyber Security* [47]. These guidelines provide a valuable basis for security by design as they propose standardized architecture patterns (NIST Logical Reference Model, NIST LRM) complemented with well-defined interfaces and corresponding security requirements. To study the integration of such fundamental concepts, the NIST LRM has been digitized and used to obtain particular design patterns as discussed for example in [13], [14], [48] or [15]. Moreover, the digital model has been made publicly available [8].

To support architecture *validation*, concepts for *static* and *dynamic* analysis have been investigated. In terms of static validation, simple aspects such as visual inspection were addressed by integration with the open source 3D visualization tool published by OFFIS [42], [49]. Moreover, Key Performance Indicator (KPI) based evaluation has been implemented that allows for both, definition and assessment of certain KPIs. This aspect has been studied in detail in context of privacy impact evaluation where assessment took place in two scenarios. First, from within the modelling tool and second, in external environments. For the second scenario, dataflow-graphs have been exported to an ontology based evaluation tool as discussed for example in [50], [51], or [52].

To better understand dynamic aspects, the DSSE Modelling Environment further interacts with the MOSAIK Co-Simulation environment[9] developed by OFFIS. The feasibility of this approach has been validated in two scenarios. First, in the field of Industry 4.0 (not yet published) and second, in the field of Smart Grids [39], [53]. Especially the second scenario is of relevance. To better understand emergent behaviour caused by electric vehicle charging, in this scenario models from the Smart Grid domain have been combined with models from electric vehicles. Thus, this atttempt already outlines a perspective for future System-of-Systems Engineering (SoSE).

Another research item addressed is the integration with design models where two scenarios can be separated. In case of non-software artefacts, a break in the toolchain appears inevitable. Thus, focus is put on traceability which is mainly addressed in context of the process model (e.g., clear definition of hand-over process and artefacts).

More benefits, however, are expected in terms of software artefacts. Assuming design models created in UML and the utilization of Round-Trip Engineering (RTE), a seamless integration appears feasible. In presence of a suitable runtime environment with integrated modules such as communication libraries or security mechanisms, more value is achievable. For example, integration of certain functionality on architectural level could be used to automatically obtain a particular configuration of the runtime environment.

In this context, our research just scratches the surface. However, first proof of concepts have been realized so far and are discussed for example in [14], [15], [23], [24]. An initial version of our OSGI based runtime environment FREDOSAR[10] has been made publicly available[11] as well.

IV. REVIEW OF THE DSSE APPROACH

As outlined in the previous sections, research on DSSE is going on for several years and various aspects were studied. In the following sections, experiences made so far are summarized in reference to different aspects.

A. *Reference Architecture Frameworks*

Work on the DSSE Approach has been triggered by the observation of the value delivered through commonly agreed frameworks such as the SGAM or RAMI. Despite this appealing aspect it has to be taken into account that not all frameworks are created with focus on engineering. Therefore, different adoptions need to be considered.

First of all, consolidation in terms of structure is required as for example *viewpoints*, *model kinds*, and their relation should follow the concepts from ISO42010 [18]. Further, ambiguous terminology (e.g., Use Case, Function, Interface,...) needs clarification.

Another aspect is the integration of requirements and cross-cutting concerns which ensures proper treatment by following well accepted concepts. This cannot be achieved by simply integrating a requirements viewpoint. Rather, guidance is needed in terms of development logic and for relation with functions, Use Cases, or components.

Asides the conceptual aspects, a thorough validation of proposed reference architectures is deemed crucial. In the Smart Grid, for example, the SGAM works out pretty well by relying on approved concepts. In contrast, RAMI introduces new concepts such as the decomposition of the problem domain into *type* and *instance*. Even though that appears logic in a first step, it turns out tricky when it comes to modelling of particular architectures [32]–[34].

---

[8] www.en-trust.at/NISTIR/

[9] https://mosaik.offis.de

[10] Free Educational Open System Architecture

[11] www.fredosar.org

Generally spoken, the existence of agreed reference architectures is of great value but always should be validated in respect to suitability for engineering.

B. Architecture and Function Modelling

System decomposition on architectural level is daily business for many architects and appears to be a straightforward task. The need for structured approaches with consistent viewpoints, closely related to a process model, is evident though. In our research, different approaches such as SYSMOD [54] or Twin-Peaks [55], [56] have been evaluated. Best results were achieved by utilization of the Software Platform Embedded Systems (SPES) methodology which provides explicit viewpoints for requirements, functions, logical, and technical aspects. This statement, however, is made on basis of our subjective experience and not backed by empirical analysis.

Another concern is the question of *function development* as the term "function" is used in ambiguous ways. Especially between engineers from hardware and software the interpretation varies. In software, for example, a function is clearly assigned to the solution domain whereas in hardware a function is considered somewhere between problem space and solution. In particular, on basis of given inputs (e.g., Use Cases) a function is engineered by *function developers* who design aspects such as interference, disturbance, chain-of-effects, and others before a hand-over to solution engineers takes place.

The discrepancy outlined here is a common source for inconsistencies and highlights the need for a clear, interdisciplinary description of function development considering the needs of all involved disciplines. In our research, we ended up using the FAS methodology [22] as suitable concept. However, the concept of function development is not adequately addressed in today's modelling languages such as SysML which makes individual adoptions necessary.

C. Stakeholder Participation

To foster active participation of different stakeholders it is not sufficient to only consider technical aspects such as viewpoints and model kinds. Also, the human factor needs to be addressed. During application of DSSE it came clear that utilization of modelling tools and especially learning their usage is a major impediment.

Good experiences have been made with a staged approach separating *concept* and *tools*. Thus, in the first stage, stakeholders participated as observers without the need for learning new tools. Therefore, models have been structured in a website like manner with integrated navigation elements. Further, these models were exported as HTML files which enabled access via a common web browser. The publicly available NIST LRM as discussed in Section III can serve as example for this concept.

The idea of low-barrier access helped raising acceptance significantly. A drawback, however, is the effort to organize models with focus on *browsability*. This aspect is addressed in our present work by integration of suitable templates on the one hand and implementation of functionality, such as automated insertion of navigation elements, on the other.

When familiar with the overall concept, during the second stage stakeholders can learn to participate in modelling. This step, however, requires willingness and adequate training. For sure, not everybody will participate actively in this task. Rather, the sweet spot for all stakeholders needs to be determined individually. A suitable concept, for example, could be to use the model export as top-level perspective that links related documents (e.g., stakeholder input) wherever useful.

Besides conducted work it is necessary to acknowledge that application of MBSE goes beyond usage of tools. Rather, the underlying mindset needs to be fully anticipated which requires to establish an accompanying change process.

D. Tool-Chain Interoperability

A central aspect for consideration is the existing lack of tool compatibility. Initially, we expected issues mainly on the vertical boundaries, especially between architectural and design models. This break in the toolchain, however, has been pretty well accepted as it implies a hand-over between different engineers and can be addressed by the process model.

A more significant obstacle showed up within individual layers, particularly within the architectural layer. Modelling tools are well suited for requirements *development*, yet limitations exist when it comes to *management* of huge requirements sets. For this reason, it is common practice to use different tools for these tasks. Gaps in standardization and varying implementation by different vendors, however, limit interoperability between *modelling* and *requirements management* tools. This aspect compromises traceability in a serious way and turned out as major barrier in industrial application.

Today, standardization bodies are already aware of this issue. Therefore, interoperability is a major concern in the ongoing standardization process for SysML V2 [57].

E. Toolbox Implementation

The availability of an easy-to use modelling environment is crucial. Our early implementations, however, fell short in achieving this goal. Focusing on technology, necessary elements were spread all over the tool. The resulting poor acceptance has been addressed by a shift towards a user-centric perspective. Therefore, instead of just listing all available functions, the implemented GUI now is structured according to the process model.

The shifted focus contributed a major step in terms of stakeholder acceptance. Implementation, however, still remains on a research level and for application in industrial scenarios significant effort remains necessary.

F. Meta DSL for Customization

The initial intention of the DSSE approach was to establish holistic concepts covering the needs of a particular domain. Application in different case studies, however, made clear that certain customization capabilities are mandatory. E.g., adoptions of the process model, the definition of specific model kinds, or integration of product families.

To address this aspect, our approach requires reconsideration. Today, the generic DSSE Approach is tailored and implemented for every considered domain. In the future, instead of providing setups for individual domains, we rather focus on implementing a so called *Meta DSL* in combination with a toolbox generator.

The idea of this approach is that users can tailor a process model according to their needs. Further, particular outcomes can be correlated with certain viewpoints, model kinds, and model elements defined on their own. Also, the specification of required functionality such as model transformations will be possible. In a second step, the user-created specification will serve as basis for instantiation of an individual toolbox out of the existing DSSE Modelling environment.

For the envisioned concept, first proof-of-concepts were implemented and the results raise the impression of feasibility. Besides plain implementation of a generic toolbox, however, more considerations are necessary on the concept. In particular, the challenge of finding the right balance between *common concepts* and *customization* needs to be addressed.

G. System-of-Systems Perspective

Until now, different domains have been considered individually. With ongoing integration of systems into System-of-Systems (SoS), more holistic considerations are necessary. For example, the integration of electric vehicles into the Smart Grid raises the need for *interoperability* and *compatibility* between models from different domains. For this purpose, concepts from different domains need to be studied to identify the common share as basis for compatibility.

In this context, a first case study has been implemented that considers integration of models from the Automotive and the Smart Grid domain. As described in [32], this case study was not limited to interoperability aspects. Moreover, it also demonstrated feasibility for exploiting these models to feed a Co-Simulation scenario, capable of identifying emergent behaviour caused by flexible energy tariffs.

Despite demonstrated feasibility of the approach, more research is necessary to better identify the common share of domains in respect to the four dimensions *semantics, processes, methods,* and *models and tools*

V. CONCLUSION AND FUTURE WORK

Despite the promising concepts of MBSE for developing complex systems, application in real world scenarios remains behind expectations. Barriers identified are the lack of acceptance by stakeholders with a background different from software engineering on the one hand, and missing concepts for a holistic engineering approach on the other hand.

The Domain Specific Systems Engineering (DSSE) approach aims at addressing both of them. To raise stakeholder acceptance, a holistic approach has been developed to support the whole engineering process. Starting with analysis in an application domain's lingua franca, the DSSE approach enables consistent modelling all the way down to particular design models. Further, it is complemented with a process model and different concepts for model integration and evaluation.

Research on DSSE has started nearly a decade ago and the approach has been continuously extended. Validation took place in different application domains such as Smart Grids, Industry 4.0, Automotive Engineering, or Smart Cities. The paper at hands summarizes the fundamental concepts of DSSE, application in different application domains, and particular research aspects studied so far. Besides the summary of existing work, the main contribution of this paper is an accumulation of learnings gained so far to identify future research fields in context of DSSE.

Considering the present condition of DSSE it appears to be a promising concept to increase practical applicability of MBSE. Despite the motivating results achieved so far, DSSE can't be considered as final answer. Rather its application in different projects keeps raising new questions.

In this light our research agenda mainly focuses on two aspects. First, a consolidation of reference architecture models in different application domains is striven for. Especially, the identification of a common share is in focus which should provide future interoperability and compatibility. Having separated the common and the individual share, work on the *Meta DSL* toolbox will continue which enables users to customize certain aspects of the DSSE approach in respect to their individual project needs.

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] H. Hick, M. Bajzek, and C. Faustmann, "Definition of a system model for model-based development," *SN Applied Sciences*, vol. 1, no. 9, p. 1074, 2019.

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

[7] 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.

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

[9] INCOSE Technical Operations, "Systems engineering vision 2020, version 2.03," INCOSE, Tech. Rep., 2007.

[10] Object Management Group, "OMG Systems Modeling Language (OMG SysML) Version 1.3," Tech. Rep., 2012.

[11] ——, "MDA Guide Version 1.0.1," Tech. Rep., 2003.

[12] ——, "Model Driven Architecture (MDA) MDA Guide rev. 2.0," Object Management Group (OMG), Tech. Rep., 2014.

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

[14] 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.

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

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

[17] Object Management Group, *Software & Systems Process Engineering Meta-Model Specification Version 2.0*, Std., 2008.

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

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

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

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

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

[23] 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.

[24] M. Fischinger, N. Egger, C. Binder, and C. Neureiter, "Towards a model-centric approach for developing dependable smart grid applications," in *2019 4th International Conference on System Reliability and Safety (ICSRS)*. IEEE, 2019, pp. 1–9.

[25] 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.

[26] P. Mattle, C. Neureiter, and F. Kupzog, "Projekt SGMS – INTEGRA Übergang zu netz- und marktgeführtem Betrieb im Smart Grid," in *Proceedings of the 4th Workshop on Communications for Energy Systems*. Vienna, Austria: Austrian Electrotechnical Association, Sep 2013, pp. 44–52.

[27] 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.

[28] 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.

[29] 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.

[30] M. Meisel, S. Wilker, J. Fabini, R. Annessi, T. Zseby, M. Müllner, W. Kastner, M. Litzlbauer, W. Gawlik, and C. Neureiter, "Methodical reference architecture development progress," in *5th DA-CH+ Energy Informatics Conference in conjunction with 7th Symposium on Communications for Energy Systems (ComForEn)*, 2016, p. 40.

[31] C. Binder, O. Jöbstl, and C. Neureiter, "RAMI 4.0 - vom Konzept zur Anwendung," *Computer & Automation*, no. S3, pp. 6–9, 2018.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

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

[37] ——, *IEC 63205 ED1: Smart Cities Reference Architecture (SCRA)*, Std., 2019.

[38] 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.

[39] 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.

[40] J. Trefke, J. M. González, and M. Uslar, "Smart grid standardisation management with use cases," in *2012 IEEE International Energy Conference and Exhibition (ENERGYCON)*. IEEE, 2012, pp. 903–908.

[41] M. Uslar, M. Specht, C. Dänekas, J. Trefke, S. Rohjans, J. M. González, C. Rosinger, and R. Bleiker, *Standardization in Smart Grids*. Springer Heidelberg New York Dordrecht London, 2013.

[42] M. Gottschalk, M. Uslar, and C. Delfs, *Tool-Support – A Use Case Management Repository*. Cham: Springer International Publishing, 2017, pp. 63–69. [Online]. Available: https://doi.org/10.1007/978-3-319-49229-2_4

[43] IEC System Committee Smart Energy, IEC 62559-3 Use case methodology - Part 3: Definition of use case template artefacts into an XML serialized format. International Electrotechnical Commission.

[44] G. Eibl, D. Engel, and C. Neureiter, "Privacy-Relevant Smart Metering Use Cases," in *Proc. International Conference on Industrial Technology (ICIT)*. IEEE, 2015, pp. 1387–1392.

[45] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic Concepts and Taxonomy of Dependable and Secure Computing," *IEEE Transactions on Dependable and Secure Computing*, vol. Vol. 1 No., pp. 11–33, 2004.

[46] C. Neureiter, G. Eibl, A. Veichtlbauer, and D. Engel, "Towards a Framework for Engineering Smart-Grid-Specific Privacy Requirements," in *Proc. IEEE IECON 2013, Special Session on Energy Informatics*. Vienna, Austria: IEEE, 2013, pp. 4803–4808.

[47] The Smart Grid Interoperability Panel – Cyber Security Working Group, "NISTIR 7628 - Guidelines for Smart Grid Cyber Security vol. 1-3. Revision 2," National Institute of Standards and Technology (NIST), Tech. Rep., August 2014.

[48] 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.

[49] M. Uslar, A. Göring, R. Heidel, C. Neureiter, D. Engel, and S. Schulte, "An Open Source 3D Visualization for the RAMI 4.0 Reference Model," in *Proceedings on VDE Kongress 2016*, Nov. 2016, pp. 1–6.

[50] 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.

[51] ——, "Model-driven privacy assessment in the smart grid," in *Proceedings of the 1st International Conference on Information Systems Security and Privacy (ICISSP)*, Feb 2015, pp. 173–181.

[52] ——, "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.

[53] 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.

[54] T. Weilkiens, *SYSMOD - The Systems Modeling Toolbox*. MBSE4U - Tim Weilkiens, 2016.

[55] B. Nuseibeh, "Weaving together requirements and architectures," *Computer*, vol. 34, no. 3, pp. 115–119, 2001.

[56] J. Cleland-Huang, R. S. Hanmer, S. Supakkul, and M. Mirakhorli, "The twin peaks of requirements and architecture," *IEEE Software*, vol. 30, no. 2, pp. 24–29, 2013.

[57] Object Management Group, "Systems Modeling Language (SysML) V2 API and Services Request For Proposal (RFP) ," Object Management Group, Tech. Rep.