

RESEARCH

Open Access



# Towards automated engineering and validation of cyber-physical energy systems

Filip Prörtl Andrén<sup>1\*</sup>, Thomas I. Strasser<sup>1</sup> , Jürgen Resch<sup>2</sup>, Bernhard Schuiki<sup>2</sup>, Sebastian Schöndorfer<sup>2</sup>, Georg Panholzer<sup>3</sup> and Christof Brandauer<sup>3</sup>

From The 8th DACH+ Conference on Energy Informatics, Salzburg, Austria. 26-27 September, 2019

\*Correspondence:

[filip.proestl-andren@ait.ac.at](mailto:filip.proestl-andren@ait.ac.at)

<sup>1</sup>AIT Austrian Institute of Technology—Electric Energy Systems, Center for Energy, Giefinggasse 2 1210, Vienna, Austria  
Full list of author information is available at the end of the article

## Abstract

The massive deployment of distributed generators from renewable sources in recent years has led to a fundamental paradigm change in terms of planning and operation of the electric power system. The usage of advanced automation and information and communication technology is a key element to handle these new challenges and to turn the traditional power system into a smart grid. The implementation of such complex systems solutions is associated with increasing development complexity resulting in increased engineering costs. The traditional engineering methods used for power system automation were not intended to be used for applications of this scale and complexity. However, the usage of proper methods, automation architectures, and corresponding tools holds huge optimization potential for the engineering process. Therefore, this work presents a model-based engineering and validation support system, covering the overall engineering process for smart grid applications.

**Keywords:** Cyber-physical energy system, Automated engineering process, Validation support system, Domain-specific language

## Introduction

The massive deployment of Distributed Energy Sources (DERs) in recent years has led to a paradigm change in terms of planning and operation of the distribution system (Liserre et al. 2010). Automation and control systems, using advanced Information and Communication Technology (ICT), are key elements to handle these new challenges and to turn the traditional power system into a smart grid (Farhangi 2010; Gungor et al. 2011; Strasser et al. 2015; Yu and et al 2016). Beyond purely technical solutions, changes in regulations and grid codes will also be indispensable.

With new approaches and concepts also new challenges emerge. The implementation of complex systems solutions is associated with increasing development complexity resulting in increased engineering costs. The traditional engineering methods used for power system automation were not intended to be used for applications of this scale and complexity (Uslar et al. 2019; Zanabria et al. 2018). However, the usage of proper methods, automation architectures, and corresponding tools holds huge optimization potential for the engineering process (Morris et al. 2009; Prörtl Andrén et al. 2017).

One methodology that can be used to reduce the engineering effort is to start with detailed use case and requirements engineering. Recent smart grid projects are also utilizing this approach for the development of different applications. Currently, the two most common methods for use cases and requirements engineering are the Smart Grid Architecture Model (SGAM) and the IEC 62559 approach (also known as the IntelliGrid method) (Uslar et al. 2019; Gottschalk et al. 2017). These methods complement each other and the main aim is to provide clear and structured documentation of the use case and their requirements.

When these use case methodologies are used properly, the results are structured use case descriptions and diagrams. Since one of the main ideas is to identify problems early in the development phase, these descriptions often contain a lot of information. Although, both SGAM and IntelliGrid strive towards machine-readable representations, a disadvantage with existing approaches is that this information is only provided in a non-formal representation. Thus, it cannot be adequately used in a computerized and automated approach. Furthermore, these existing use case and design methods do not allow the reuse of other already existing input specifications that are typically provided as an input to the engineering process, such as IEC 61850 specifications or power system models (Zanabria et al. 2018).

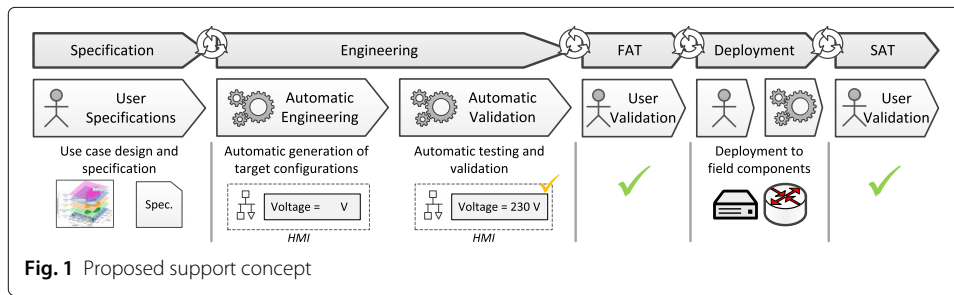
Consequently, current engineering approaches require a significant amount of avoidable manual work. Since information that has already been provided by the engineer during the use case design is not machine-readable, it cannot be used in an automated fashion in the following engineering phases (implementation, validation, and deployment). As a consequence, the same information gets “redefined” in the later phases. This is a very time-consuming and error-prone approach to engineering smart grid applications. Some more formal approaches have already been developed which are based on model-driven engineering/rapid prototyping (Uslar et al. 2019; Zanabria et al. 2018; Pröstl Andrén et al. 2017; Faruque and Ahourai 2014; Faschang et al. 2013; Neureiter 2017; Pröstl Andrén et al. 2018; Pröstl Andrén 2018) and test-driven design (Zanabria et al. 2018; Blank et al. 2016; Strasser et al. 2017) but a comprehensive framework which also includes the development and validation of visualisation concepts is missing until now.

This paper addresses these issues with a concept for an automated and model-based engineering and validation framework which is currently being developed in the Austrian research project “MESSE”. It covers the whole development process of smart grid automation applications and provides development support for specification and design, engineering, validation, and finally deployment and operation for smart grid applications (incl. automation/control, communication, and visualization).

The remainder of the paper is organized as follows: The first section provides a brief overview of the support concept idea whereas the engineering and validation chain is shown afterwards. A first prototypical realization is outlined in the following section. Conclusions and future work are discussed in the final section.

### **Support concept overview**

This work proposes a concept for a model-based engineering and validation support system, covering the overall engineering process for smart grid applications—from use case design to validation, and finally deployment and commissioning. Based on a model-driven development approach, the methodology consists of multiple parts as depicted in Fig. 1.

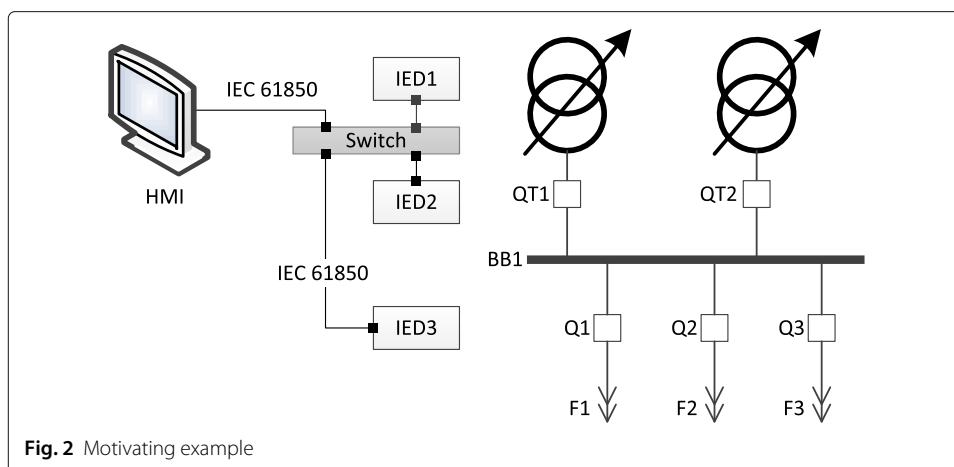


This paper focuses on: (i) specification and use case design, (ii) automated engineering, and (iii) automated validation.

To illustrate how the engineering support concept is used, a motivating example is provided in Fig. 2. The goal is to generate the visualization (i.e., Human Machine Interface/HMI) based on the power system one-line-diagram and the communication setup.

The first part of the engineering support concept is using a formal modelling approach for specification and use case analysis. During this part, existing specifications are collected and finalized by the user. In the example, this includes ICD files that describe the IEC 61850 capabilities of the IEDs. Furthermore, an HMI visualization template is collected from the utility operator, specifying the layout and the general behaviour of the HMI. The specification and design approach is based on the Power System Automation Language (PSAL) (Prörtl Andrén et al. 2017). PSAL is a Domain Specific Language (DSL) and a use case modelling technique based on SGAM and IntelliGrid (Uslar et al. 2019). A more detailed introduction of PSAL is given below.

The second part of the support concept starts with an automatic generation of target configurations. This can be automatically generated HMIs and visualizations configured for use in SCADA systems and it can also be target configurations for ICT and network components. They can be used on the one side for communication protocol configurations (e.g., IEC 61850 configurations), but also for low-level configuration of the communication network (e.g., configuration of Software Defined Networking (SDN) controllers (McKeown et al. 2008; Gopalakrishnan 2011)). In the example, the HMI is



generated, containing a visualization of the one-line-diagram and configured interfaces to the IEDs.

After the target configurations have been generated the next step is to validate them using an automated testing approach. One basis for these tests can be validation scenarios and test specifications, accompanied by expected test results, as provided by the user. However, based on the formal specification it is also possible to implicitly derive certain tests, such as reachability between IEDs within a network. The validation can on the one side be done using pure software tests, but can also be realized in combinations of software, hardware and simulations. If hardware is involved, manual setup—according to generated setup guidelines—may be needed. The validation also includes a human confirmation. Preferably this is only needed for specific acceptance tests (e.g., Factory Acceptance Test (FAT), Site Acceptance Test (SAT)). In the example, a FAT would include a validation that the HMI can reach each IED, that the data points are accessible, and that changes in the data points are visualized in the HMI accordingly.

### **Automated engineering and validation chain**

The approach and design that is chosen to structure and finally implement the automated engineering and validation chain is depicted in Fig. 3. In the specification phase, the user creates, collects, and completes various input specifications and consolidates them into a common model. The user can thereby be supported in multiple ways by an integrated development environment (IDE) providing advanced editing support but also import functions for standardized formats as well as wizards that guide a semi-automated enrichment of the model. The model, which is formalized in PSAL (see below), serves as the starting point for the subsequent generators which mark the beginning of the automated engineering process.

Multiple generators transform the model to target configurations for various aspects of a power system. There are generators for functions, communication networks, IED configurations, HMIs and validation tests. The generation processes are parameterized by so called *generator models* ( $xGM$  in Fig. 3). They contain all the information and resources that are specific to the generation process and which shall thus not be contained in the model itself. A generator model may contain configuration parameters, resources (e.g. symbol libraries), and executable code.

Typically, a generator model is customer specific. Also, different generator models will be used over the course of the engineering process. Our design explicitly supports an iterative engineering process by separating the target-specific aspects of the generation process from the model itself. For example, the user might initially generate configurations targeted at a purely simulated/virtualized target environment. In a later iteration, the user can move on to a combined setup of software / hardware based components by simply employing a different generator model.

A generated target configuration is deployed to the respective target by a configuration tool (not shown in Fig. 3). This step is parameterized by a configuration tool model, which contains, e.g., the credentials that are required to access an IED or a networking component of the target environment, respectively. Besides configuring a target, a configuration tool can also produce setup guidelines that are required for a manual setup (e.g. a wiring plan).

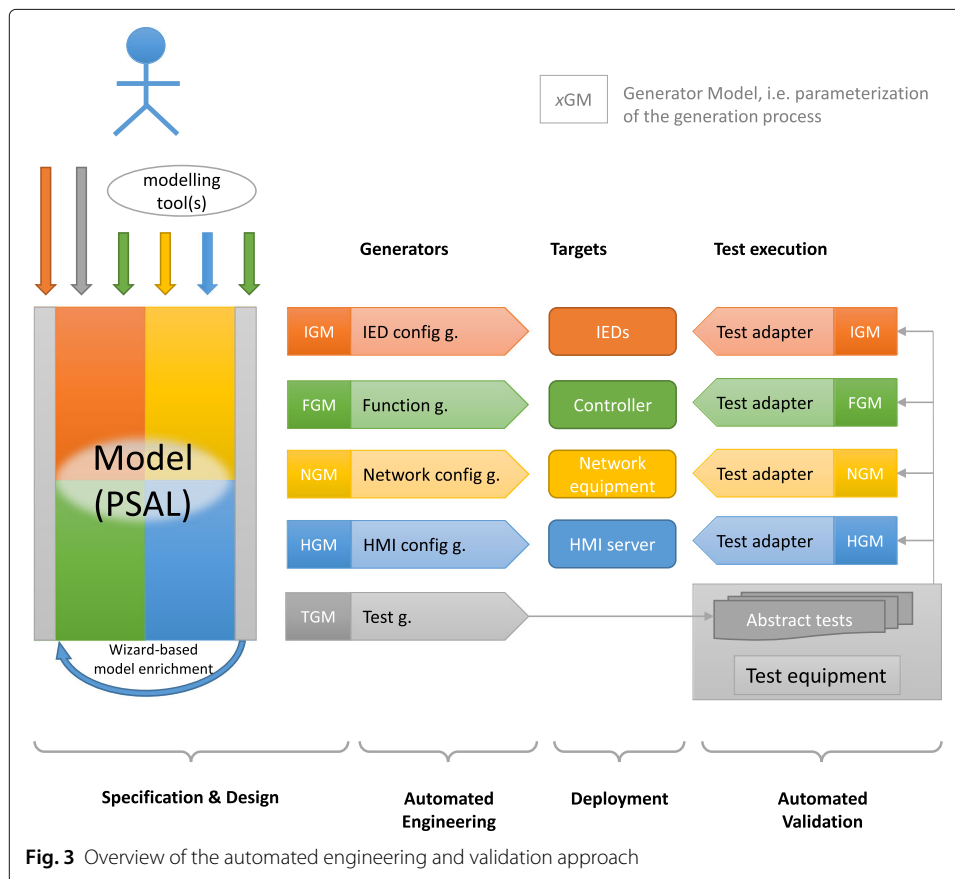


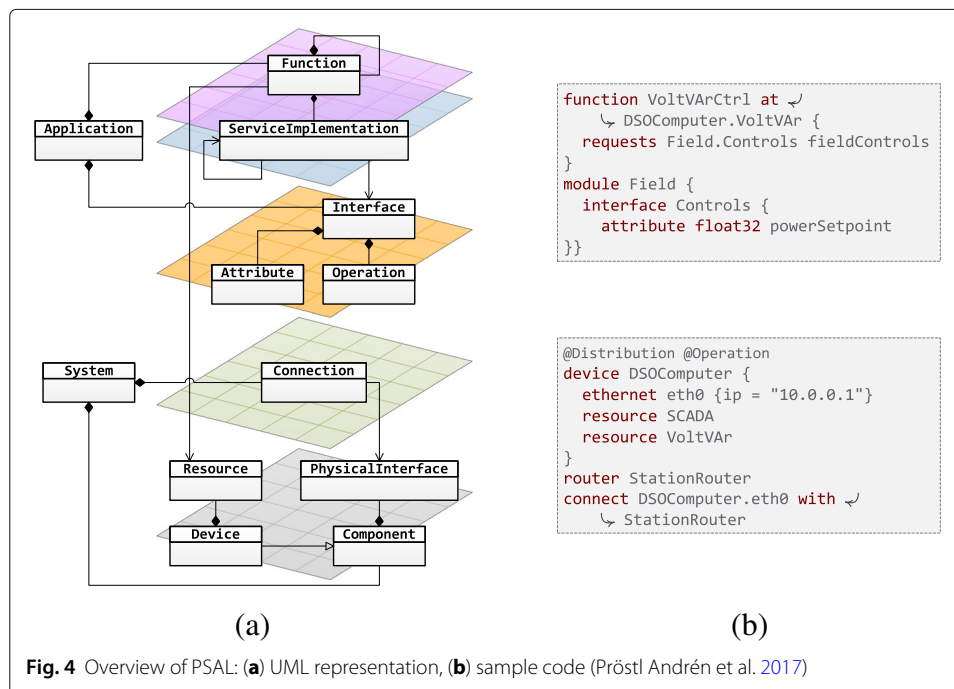
Fig. 3 Overview of the automated engineering and validation approach

As can be seen in Fig. 3, our design also includes a test generator which generates validation tests. We note that based on the rich formal model provided in the specification phase it is possible to derive many tests automatically without the need for a user to define validation scenarios in the specification phase. We refer to such a test as an *implicit* test as the test logic and the way to determine the test verdict is implicitly defined. Implicit tests may seem somewhat trivial, however, industry practice shows that they account for a large share of engineering costs (e.g. in the testing of the communication from an IED to an HMI) that can be drastically reduced (MacDonald 2018). Implicit tests can, e.g., verify that i) devices and functions are indeed able to communicate with each other; ii) signals are indeed correctly transmitted and displayed in the HMI; iii) alarm thresholds are respected.

### Extended model-based specification and design

The specification and design part (see Fig. 3) is mainly based on PSAL (Prörtl Andrén et al. 2017). PSAL's intention is to provide a formalized language for SGAM compatible use case design and at the same allowing rapid development of automation, control, and ICT functions of power system applications (Andrén et al. 2016). Thus, PSAL supports the development of high-level use case descriptions, combined with more detailed use case specifications.

PSAL is a Domain Specific Language (DSL) that allows the user to model power system use cases. It supports design of all SGAM layers using a textual language. In Fig. 4, a UML



representation of PSAL's metamodel as well as example implementations of an *Application* and a *System* are shown. One of the main ideas with PSAL is that it should allow rapid generation of code and configurations (Prösl Andrén et al. 2017).

The original version of PSAL mainly focused on how to describe functions and the communication between these functions. Also, the generation of communication configurations such as IEC 61850 or Modbus was supported. Although these configurations can be used to set up a working communication between two IEDs, they are not enough to generate the visualization part of an HMI since the information regarding the system layout is missing.

Especially for HMIs, there is a currently ongoing standardization activity that is focusing on extending IEC 61850-6 with graphical requirements for an HMI application. The final result of this work will be IEC 61850-6-2 ("Configuration description language for extensions for human machine interfaces"). Using this new approach, the integration between the IEC 61850 model and the HMI tools will be further facilitated and improved. For example, with the new IEC 61850-6-2 part it will be possible to provide HMI configurations—also including visualization information—in a tool-independent way (Prösl Andrén et al. 2018). In order to utilize this ongoing activity, PSAL was extended to increase the compatibility with IEC 61850. An overview of the extended mapping between PSAL and IEC 61850 is shown in Table 1.

Using this new extended mapping it is now possible to generate the necessary information from PSAL to completely implement a fully functional HMI: visualization based on the one-line-diagram representation and the configurations needed to set up the communication from the HMI to the IEDs. Apart from this, PSAL already offers a range of descriptions mechanisms that can be used for validation and testing purposes (Prösl Andrén 2018).

**Table 1** Mapping parts of the IEC 61850 substation section into PSAL parts

IEC 61850	PSAL Object	PSAL Example Code
tSubstation	System	<code>system</code> SystemName {}
tVoltageLevel	VoltageLevel	<code>voltagelevel</code> VoltageLevelName {}
tBay	Bay	<code>bay</code> BayName {}
tPowerTransformer	Transformer	<code>transformer</code> TransformerName {}
tTransformerWinding	TransformerWinding	<code>winding</code> WindingName {}
tConductingEquipment	ElectricalEquipment	<code>generator</code> GeneratorName {}
tTerminal	Terminal	<code>terminal</code> TerminalName
tConnectivityNode	Connection	<code>connect</code> TerminalName <code>with</code> OtherTerminal

### Generation of network configurations

PSAL provides capabilities to model network interfaces, communication links and switches that make up the network topology. Based on this model it is possible to automatically generate a low-level network configuration that can be turned into an operational network in various target environments. Auto-generated communication networks are particularly interesting in the context of automating validation tests. The goal is to completely automate the generation of a fully functional operative network that can be employed in validation tests.

In PSAL, all physical connections between devices as well as logical connections between functions on the application layer are modelled. This allows to infer which devices need to be able to communicate with each other and which don't. It is consequently possible to generate a VLAN configuration that is tailored to those exact communication requirements. This is typically not done today in industry as it is a complex task when done manually. Also, the effort for testing the communication network increases. In our concept, however, the generation and testing of such a restrictive, security-enhancing communication configuration comes at no additional manual cost.

As mentioned above, a key concept is to support various different targets like e.g. i) network simulation/emulation; ii) virtualized networks; iii) SDN-based configurations of real hardware. To this end, the generation of network configurations is designed as a 2-step process: in the first step, the PSAL model is mapped to a target-agnostic model of a communication network. The transformation to a target specific configuration is done in a second step. A network configuration tool can then deploy the configuration to a given target.

### Generation of an HMI

The creation of an HMI is typically based on existing description files that model the underlying system. Due to the widespread use of IEC 61850, the development of an HMI is nowadays often based on *Substation Configuration Description* files (IEC 61850-6) from which the voltage levels, bays and additional information can be extracted. There are, however, several essential aspects on an HMI that can currently not be modelled in IEC 61850-6, e.g.: i) the overall appearance, navigation and interaction offered by the HMI; ii) the shape and color of HMI elements; iii) alarm lists. The upcoming standard IEC 61850-6-2 is expected to remedy the situation but then again not all projects are based on IEC 61850.



A higher-level modelling language like PSAL affords the opportunity to provide HMI-related model elements. Consequently, a unified and consistent model containing the layout, logical connections of the primary, secondary and ICT equipment as well as HMI aspects can be formulated in a protocol-independent way. Again, as described above, the customer- and target-specific aspects of the HMI are not included in the PSAL model but they are instead separated into the generator model. It includes, e.g., symbol libraries, color schemes, etc.

The modelling environment can support the engineer in various ways. A graphical wizard can, e.g., match the communication protocols that are supported by the IEDs (as defined in the IED capability files collected in the specification phase) with the ones that are supported by the HMI and enable a selection with a few clicks.

Once triggered, the HMI generator – parameterized by a generator model – maps the PSAL model to an executable HMI target configuration comprising the single line diagram, the configuration of the IED clients, and the necessary signal data points.

### Generation of validation tests

Each and every signal that has a representation in the HMI has to be tested. The validation has to be done in an “end-to-end” test, i.e. it must include the complete processing chain starting with the stimulus at the IED, including the network transmission, and ending with the modified HMI rendering (e.g. change of pixels). In addition, for every control command from the HMI, it has to be tested whether it stimulates the correct outputs on the IED. A signal test usually requires the use of a special test equipment to trigger inputs of the IED. On the HMI side, a special testing API provides an interface to detect the reaction of the HMI, e.g. a new entry in the event list or the change of the shape of a switch gear representation. For the control tests, a control command is triggered via the HMI testing API and it needs to be verified if the expected output occurs at the IED.

Communication related tests need to validate that (only) connectivity on the device and application level is indeed available as specified in the PSAL model.

As described above, our approach comprises a test generator that creates validation tests based on the specifications formalized in the PSAL model. The generated tests are called *abstract* tests as they require a test execution environment as well as implementations of the so called *test adapters* to become executable. These adapters are an important building block of our testing concept. A test adapter provides a target-agnostic API that defines the basic testing capabilities/functionalities upon which an abstract test can be built. Internally, a test adapter provides an implementation of this API for a specific target system. This approach follows proven industry practice, see, e.g., the *System Adapter* in TTCN-3 (Methods for Testing and Specification (MTS) 2017).

### Prototypical realization and validation

PSAL has been implemented using the xText (Efftinge and Völter 2006) environment for the Eclipse IDE (Prössl Andrén 2018). The specification and design is carried out by the engineer using a PSAL editor. The editor supports the user by providing syntax highlighting, code completion, and validation of the syntax as well as the semantics of PSAL. Listing 1 shows an excerpt of the PSAL code describing the motivating example from Fig. 2.



```

1 system S1 {
2   transformer T {
3     winding W1 { terminal t }
4     winding W2 { terminal t }
5   }
6   voltagelevel VL-400V {
7     bay B1 {
8       breaker QB1 {
9         terminal t1
10        terminal t2
11      }
12      connect T.W2.t with QB1.t1
13      ...
14    }
15  }
16  device IEDTransformer {
17    ethernet eth0
18    resource CTRL
19  }
20  device IEDBay1 {
21    ethernet eth0
22    resource CTRL
23  }
24  switch SubstationSwitch
25  connect IEDTransformer.eth0 with
    SubstationSwitch
26  connect IEDBay1.eth0 with SubstationSwitch
27  ...
28 }

```

**Listing 1** Excerpt of the PSAL code for the motivating example from Fig. 2

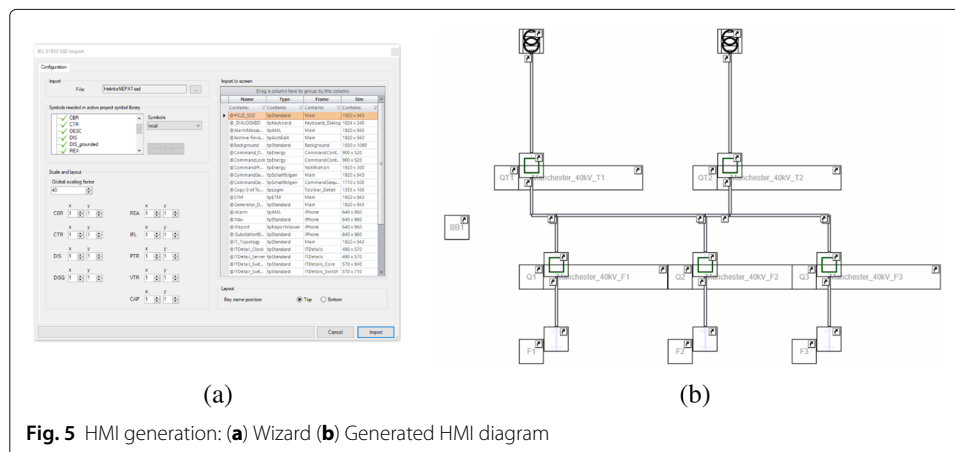
As an example of semi-automated model enrichment we implemented an Eclipse wizard that can be used to automatically generate IP addresses for ICT devices. This can be useful in an early iteration phase where the engineer does not want to deal with details such as IP addresses (which will possibly change in a later iteration, anyway) but these addresses are nevertheless needed for many tests (e.g. to configure the drivers in HMI tests).

In order to map PSAL to IEC 61850, as proposed in Table 1, the Atlas Transformation Language<sup>1</sup> (ATL) was used. A model-to-model transformation from PSAL to IEC 61850 was defined in ATL. When executed, the transformation creates an IEC 61850 SCL description based on the information provided in the PSAL model.

The generation of a communication network is implemented as a 2-step process. For the first step, we implemented a Java program that transforms the PSAL document into a generic model of a communication network. For the second, target-specific step, we wrote a python script that uses this generic network model to instantiate a virtualized communication network based on Mininet (Lantz et al. 2010). Mininet is a network emulator that runs virtualized end-systems, switches, routers, links, and controllers using a lightweight virtualization technology. These virtualized components run the same kernel, networking stack, and applications as the underlying Linux-based host operating system. Through the Mininet API we can setup an arbitrary topology (including bandwidth restrictions, emulated delays, and artificial packet losses if needed) in a fully automated fashion within seconds. It is possible to provide specific docker images for the virtualized hosts (e.g. to provide a software-based IED for testing). The virtual hosts can communicate with real hosts that are connected via a real network to the machine running Mininet. The prototype can easily run a complete virtualized version of a substation network on a single PC which is connected to a real HMI server.

The generation of the HMI configuration is currently also realized as a 2-step process. In this case, the PSAL document is first mapped to an SCL file (IEC 61850) plus additional proprietary configuration files. The step may seem questionable as we outlined above that it is currently not possible to model several HMI-related aspects in SCL. However, due to the wide-spread use of IEC 61850 and its high economic relevance, an SCL-based HMI generation can be applied directly in current industrial use-cases. Indeed, as a pre-cursor to the current project, we have implemented a proof-of-concept wizard that generates an HMI from an SCL file plus additional sources of information. In the current prototype, an extended version of this wizard is used to implement the second step. Figure 5 shows a

<sup>1</sup><https://www.eclipse.org/at/>



screenshot from the wizard (left) as well as the single line diagram of the generated HMI (right).

The work on the automated generation of validation tests has only just started in the project. At the moment the prototype merely generates some implicit tests related to the communication network. For every network link (i.e., a connect between two devices in PSAL – see sample code in Fig. 4) we run connectivity tests between the two connected devices in the generated communication network. Furthermore, we run connectivity tests for all the generated VLANs. Each device needs to be able to reach all the other devices in the same VLANs but must not reach any other device. Although these tests might seem somewhat superfluous they are an indispensable part of a comprehensive validation test suite. Even for a correctly auto-generated network they will uncover certain configuration issues, e.g., IP mismatches, and wiring errors (if manual wiring is involved).

## Conclusions and future work

This paper presents a concept of an automated, model-based engineering and validation framework that covers the overall development process of smart grid applications, from specification and use case design, to automatic engineering and validation, and finally deployment and commissioning. The concept is rooted in a domain specific modelling language (PSAL) for SGAM compatible use case design. It enables an engineer to model power system use cases, develop automation, control, and ICT functions and it is specifically designed to facilitate code generation.

We developed an approach to an engineering chain that implements an automated generation of target configurations as well as validation tests from a PSAL model. Our concept encourages an agile and iterative development model. This is achieved by a) clearly separating the PSAL model from customer- and target-specific generation options and b) supporting the generation of various target configurations.

The project is now entering the phase where the main focus is put on the automated generation of validation tests. We will exploit the concept of target-specific test adapters that provide target-agnostic APIs. These APIs provide the basic building blocks upon which abstract tests can be formulated. We see a large potential for validation tests that can be derived from the PSAL model in a fully automated fashion. Overall, the results established so far suggest that our engineering concept can automate many steps that are

traditionally carried out manually. We thus expect our approach to afford similar benefits as have been achieved in other domains (Drath et al. 2008; Li et al. 2019; Hästbacka et al. 2011; Markthaler et al. 2017).

In the upcoming validation phase we will study if the expected benefits of our automated engineering and validation concept can indeed be attained. Our main validation scenario is focused on the engineering and testing of a substation HMI. We selected this scenario as our long-standing experience in this field allows us to evaluate how the usage of our new concept stacks up against the traditional approach. We know from many completed projects how much time it takes on average to develop a substation HMI. An even larger effort is needed to test the communication configuration from IEDs to the HMI. Typically every data point has to be tested from its entrance into the IED up to the HMI to guarantee its correct configuration. External test equipment needs to be installed to simulate the signals on the input terminals. The appearance of the signal in the HMI has to be checked manually. This is often a very time-consuming job considering that there can be up to 7000 signals to be tested. With the validation of our approach, we believe the evaluation will show that an automated approach can significantly reduce the amount of time needed for development and testing.

#### Authors' contributions

All authors contributed to the concept; FPA, GP, CB, BS, and SS contributed to the prototypical realization. All authors contributed to the writing of the paper and they have read and approved the final manuscript.

#### Funding

This work received funding from the Austrian Ministry for Transport, Innovation and Technology (bmvit) and the Austrian Research Promotion Agency (FFG) under the ICT of the Future Program in the MESSE project (FFG No. 861265). Publication of this supplement was funded by Austrian Federal Ministry for Transport, Innovation and Technology.

#### Availability of data and materials

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

#### About this supplement

This article has been published as part of *Energy Informatics* Volume 2 Supplement 1, 2019: Proceedings of the 8th DACH+ Conference on Energy Informatics. The full contents of the supplement are available online at <https://energyinformatics.springeropen.com/articles/supplements/volume-2-supplement-1>

#### Competing interests

The authors declare that they have no competing interests.

#### Author details

<sup>1</sup>AIT Austrian Institute of Technology—Electric Energy Systems, Center for Energy, Giefinggasse 2 1210, Vienna, Austria. <sup>2</sup>Ing. Punzenberger COPA-DATA GmbH, Karolingerstraße 7b 5020, Salzburg, Austria. <sup>3</sup>Salzburg Research Forschungsgesellschaft m.b.H., Jakob-Haringer-Str. 5/III 5020, Salzburg, Austria.

Published: 23 September 2019

#### References

- Andrén FP, Strasser T, Kastner W (2016) Applying the SGAM methodology for rapid prototyping of smart Grid applications. In: IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society, Florence. pp 3812–3818. <https://doi.org/10.1109/IECON.2016.7794057>
- Blank M, Lehnhoff S, Heussen K, Bondy DEM, Moyo C, Strasser T (2016) Towards a foundation for holistic power system validation and testing. In: 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), Berlin. pp 1–4. <https://doi.org/10.1109/ETFA.2016.7733672>
- Drath R, Lüder A, Peschke J, Hundt L (2008) AutomationML – The Glue for Seamless Automation Engineering. In: 2008 IEEE International Conference on Emerging Technologies and Factory Automation. Hamburg. pp 616–623. <https://doi.org/10.1109/ETFA.2008.4638461>
- Efftinge S, Völter M (2006) oAW xText: A framework for textual DSLs. In: Workshop on Modeling Symposium at Eclipse Summit Vol. 32. p 118
- Farhangi H (2010) The path of the smart grid. *IEEE Power Energy Mag* 8(1):18–28

- Faruque MAA, Hourai F (2014) A model-based design of Cyber-Physical Energy Systems. In: 2014 19th Asia and South Pacific Design Automation Conference (ASP-DAC), Singapore. pp 97–104. <https://doi.org/10.1109/ASPDAC.2014.6742873>
- Faschang M, Kupzog F, Mosshammer R, Einfalt A (2013) Rapid control prototyping platform for networked smart grid systems. In: IECON 2013-39th Annual Conference of the IEEE Industrial Electronics Society. IEEE, Vienna. pp 8172–8176. <https://doi.org/10.1109/IECON.2013.6700500>
- Gopalakrishnan A (2011) Applications of software defined networks in industrial automation an architectural analysis. see [https://www.researchgate.net/publication/283441470\\_Applications\\_of\\_Software\\_Defined\\_Networks\\_in\\_Industrial\\_Automation\\_An\\_architectural\\_analysis](https://www.researchgate.net/publication/283441470_Applications_of_Software_Defined_Networks_in_Industrial_Automation_An_architectural_analysis)
- Gottschalk M, Uslar M, Delfs C (2017) The Use Case and Smart Grid Architecture Model Approach: The IEC 62559-2 Use Case Template and the SGAM Applied in Various Domains. Springer Verlag Heidelberg, Tiergartenstrasse 17, D-69121 Heidelberg
- Gungor VC, Sahin D, Kocak T, Ergut S, Buccella C, Cecati C, Hancke GP (2011) Smart Grid Technologies: Communication Technologies and Standards. *IEEE Trans Ind Informa* 7(4):529–539
- Hästbacka D, Vepsäläinen T, Kuikka S (2011) Model-driven development of industrial process control applications. *J Syst Softw* 84(7):1100–1113
- Lantz B, Heller B, McKeown N (2010) A network in a laptop: Rapid prototyping for software-defined networks. In: Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks. Hotnets-IX. ACM, New York, USA. pp 19–1196. <https://doi.org/10.1145/1868447.1868466>
- Li H, Zou M, Hogrefe G, Ryashentseva D, Sollfrank M, Koltun G, Vogel-Heuser B (2019) Application of a multi-disciplinary design approach in a mechatronic engineering toolchain. at – Automatisierungstechnik 67(3):246–269
- Liserre M, Sauter T, Hung JY (2010) Future energy systems: Integrating renewable energy sources into the smart power grid through industrial electronics. *IEEE Ind Electron Mag* 4(1):18–37
- MacDonald D (2018) Digital Substation Investment: Creating a compelling business case for investment, that balances the need for innovation with the pressure to achieve rapid ROI Presentation at see <http://digitalsubstation.com/en/2018/02/26/intellisub-europe-2018/>
- Markthaler M, et al. (2017) Improving model-based testing in automotive software engineering. In: 2018 IEEE/ACM 40th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP), Gothenburg. pp 172–180
- McKeown N, Anderson T, Balakrishnan H, Parulkar G, L. Peterson L, Rexford J, Shenker S, Turner J (2008) Openflow: Enabling innovation in campus networks. *Comput Commun Rev* 38:69–74. <https://doi.org/10.1145/1355734.1355746>
- Methods for Testing and Specification (MTS) (2017) The Testing and Test Control Notation version 3; Part 5: TTCN-3 Runtime Interface (TRI) Standard, ETSI, Sophia Antipolis Cedex, FR
- Morris TH, et al. (2009) Engineering future cyber-physical energy systems: Challenges, research needs, and roadmap. In: 41st North American Power Symposium, Starkville. pp 1–6. <https://doi.org/10.1109/NAPS.2009.5484019>
- Neureiter C (2017) A domain-specific, model driven engineering approach for systems engineering in the smart grid. PhD thesis Carl von Ossietzky Universität Oldenburg, Computer Science Department
- Prörtl Andrén F, Strasser TI, Kastner W (2017) Engineering smart grids: Applying model-driven development from use case design to deployment Vol. 10. <https://doi.org/10.3390/en10030374>
- Prörtl Andrén F (2018) Model-driven engineering for smart grid automation. PhD thesis, TU Wien
- Prörtl Andrén F, Strasser T, Resch J, Schuiki B, Brandauer C, Panholzer G (2018) Model-based Engineering and Validation Support for Cyber-Physical Energy Systems. In: Proceedings of the PAC World Conference 2018. p 7. tAC World Conference 2018, Sofia; 2018-06-26 – 2018-06-28
- Prörtl Andrén F, Strasser T, Seidl C, Resch J, Brandauer C, Panholzer G (2018) On fostering smart grid development and validation with a model-based engineering and support framework. In: Proceedings of the CIRED Workshop 2018. talk: CIRED Workshop 2018, Ljubljana. 2018-06-07 – 2018-06-08
- Strasser T, Andrén F, Kathan J, Cecati C, Buccella C, Siano P, Leitao P, Zhabelova G, Vyatkin V, Vrba P, et al (2015) A review of architectures and concepts for intelligence in future electric energy systems. *IEEE Trans Ind Electron* 62(4):2424–2438
- Strasser T, Andrén FP, Lauss G, Bründlinger R, Brunner H, Moyo C, Seidl C, Rohjans S, Lehnhoff S, Palensky P, et al (2017) Towards holistic power distribution system validation and testing—an overview and discussion of different possibilities. *e & i Elektrotechnik und Informationstechnik* 134(1):71–77
- Uslar M, Rohjans S, Neureiter C, Prörtl Andrén F, Velasquez J, Steinbrink C, Efthymiou V, Migliavacca G, Horsmanheimo S, Brunner H, Strasser TI (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). <https://doi.org/10.3390/en12020258>
- Yu X, et al (2016) Smart Grids: A Cyber-Physical Systems Perspective. *Proc IEEE* 104(5):1058–1070
- Zanabria C, Andrén FP, Strasser TI (2018) Comparing specification and design approaches for power systems applications. In: 2018 IEEE PES Transmission & Distribution Conference and Exhibition - Latin America (T&D-LA), Lima. pp 1–5. <https://doi.org/10.1109/TDC-LA.2018.8511777>

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.