Cyber-Physical Energy Systems Modeling, Test Specification, and Co-Simulation Based Testing

A. A. van der Meer*, P. Palensky*, K. Heussen†, D. E. Morales Bondy†, O. Gehrke†, C. Steinbrink‡,
M. Blank‡, S. Lehnhoff‡, E. Widl§, C. Moyo§, T. I. Strasser§, V. H. Nguyen¶, N. Akroud‖, M. H. Syed**,
A. Emhemed**, S. Rohjans††, R. Brandl‡‡ A. M. Khavari^x

*Delft University of Technology, Delft, The Netherlands, a.a.vandermeer@tudelft.nl
†Technical University of Denmark, Lyngby, Denmark
‡OFFIS e.V., Oldenburg, Germany
§AIT Austrian Institute of Technology, Vienna, Austria
¶University Grenoble Alpes, G2Elab, Grenoble, France
‖Ormazabal Corporate Technology, Bilbao, Spain
**University of Strathclyde, Glasgow, United Kingdom
††HAW Hamburg University of Applied Sciences, Hamburg, Germany
‡‡Fraunhofer Institute of Wind Energy and Energy System Technology, Kassel, Germany
^xEuropean Distributed Energy Resources Laboratories (DERlab) e.V., Kassel, Germany

Abstract—The gradual deployment of intelligent and coordinated devices in the electrical power system needs careful investigation of the interactions between the various domains involved. Especially due to the coupling between ICT and power systems a holistic approach for testing and validating is required. Taking existing (quasi-) standardised smart grid system and test specification methods as a starting point, we are developing a holistic testing and validation approach that allows a very flexible way of assessing the system level aspects by various types of experiments (including virtual, real, and mixed lab settings). This paper describes the formal holistic test case specification method and applies it to a particular co-simulation experimental setup. The various building blocks of such a simulation (i.e., FMI, mosaik, domain-specific simulation federates) are covered in more detail. The presented method addresses most modeling and specification challenges in cyber-physical energy systems and is extensible for future additions such as uncertainty quantification.

I. Introduction

The decarbonisation of the European power generation requires a high penetration of distributed, Renewable Energy Sources (RES). Their intermittent behaviour and limited storage capabilities present new challenges to power system operators in maintaining the security of supply and the power quality [1]. However, advanced Information and Communication Technologies (ICT), automation approaches, and corresponding algorithms provide new possibilities and intelligent solutions for operating power grids in a more optimized way [2], [3]. As a consequence of these developments the traditional power system is transformed into a Cyber-Physical Energy System (CPES), a smart grid [4]. Previous and ongoing research activities have mainly focused on validating certain aspects of smart grids, but until now there is no integrated approach for analysing and evaluating complex configurations in a cyber-physical systems manner available [5].

In the process of designing and developing a specific solution, validating and testing the correctness is an essential stage.

CPES like smart grids are a combination of different technologies across heterogeneous domains (power, ICT/automation, markets, customer behaviour, etc.), which have mutual interactions and inter-dependencies. Before deploying algorithms and solutions, field tests are needed to evaluate the integration on a system level, addressing all relevant domains. Up-to-now such a cyber-physical approach for designing, analysing, and validating smart grid systems is missing. The existing laboratory-based testing approaches often focus on a certain sub-system (or business sector) and its components. The integration of components—including analysis and evaluation—is not yet addressed sufficiently in a holistic manner.

Simulation-based experiments are one of the alternative testing approaches that can cover multiple domains [6]. However, the development of smart grid solutions and technologies has increased the need for a more integrated simulation approach covering all targeted areas [7], [8]. A general framework for smart grid validation and roll-out is necessary. One of the main barriers to this has been the lack of design approaches and corresponding software tools that are capable of simulating power and ICT systems holistically [5].

The lack of such system validation for smart grids is especially addressed by the European ERIGrid project [9]. By providing a Pan-European research infrastructure ERIGrid supports the technology development as well as the roll-out of smart grid solutions and concepts. It tackles a holistic, CPES-based approach by integrating European research centres and institutions with outstanding research infrastructures and jointly develops common methods, concepts, and procedures.

The aim of this paper is to discuss advanced modeling approaches, a formal specification of corresponding validation scenarios, and co-simulation based testing methods for CPES that are being developed in ERIGrid.

The rest of this paper is organized as follows: Section II

gives an overview of the proposed methodology for a holistic test description, while Section III discusses the co-simulation of CPES. Necessary software and component interfaces which are being used in the ERIGrid co-simulation environment are introduced in Section IV. A proof-of-concept example of both the test specification and the co-simulation based experiments is described in Section V. Finally, a discussion and an outlook on planned future research is provided in Section VI.

II. METHODOLOGY FOR HOLISTIC TEST DESCRIPTION

Smart grid components and functions are embedded in a distributed and complex infrastructure. The formulation and specification of test requirements is therefore both a technical and conceptual challenge. On the one hand, the technical aspect of carrying out integrated experiments by means of real-time, hardware-in-the-loop or co-simulation technologies has received significant attention in recent years [8], [10]. On the other hand, significant conceptual development and consolidation has been achieved in the ICT aspects of smart grids through standardization and harmonization of use case descriptions, system architecture, and interoperability test specification [11], [12], [13], [14]. These technical and conceptual developments have been largely pursued in independent tracks. A wide gap remains in formulating test specifications that facilitate the holistic evaluation of the integrated cyber-physical system. The initial step towards carrying out holistic tests of CPES is to harmonize the concepts and testing methods used by Research Infrastructures (RIs). Further steps then entail the harmonization of the test evaluation methodology, the harmonization of RI internal and external interfaces, as as well as interchangeable configuration descriptions. This will be of particular importance for managing the high complexity of experiments conducted across multiple RIs.

In order to frame this approach we propose the following definition of **holistic testing**: *"The process and methodology for evaluation of a concrete function, system or component within its relevant operational context with reference to a given test objective."* A challenge is therefore to formalize the complete cyber-physical system context and test criteria in a common framework. A first outline of the holistic testing approach was presented in [15], where a focus was placed on the concept of holistic testing and a corresponding methodology. The challenge of addressing multiple RIs with a single test case, as outlined in [15], is within the scope of the presented approach, but details will be omitted for brevity.

A. Incremental Test Description

Based on these concepts, we have formulated a procedure for holistic test specification, which defines steps for formulating a *concrete* and *holistic* test description (cf. Fig. 1, steps ①-④). In this process, we separate the questions "what needs to be tested?" and "why does it need testing?", which are answered in step ①, from the question "how should it be tested?", which is answered in step ③. The above questions can be answered independent of the respective testing infras-
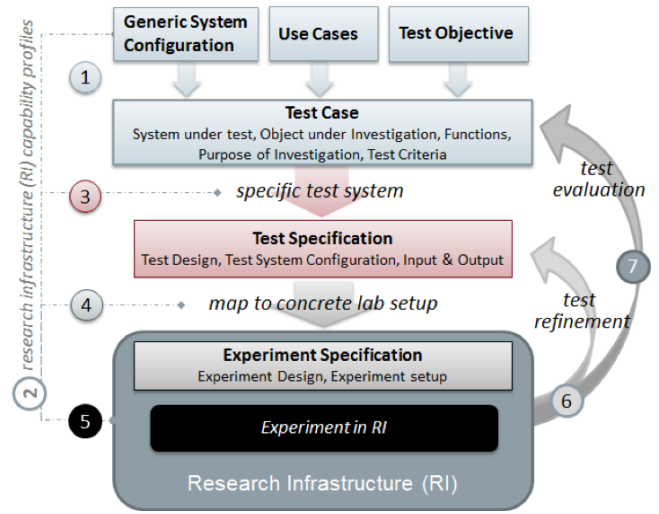

Fig. 1. The main steps in the ERIGrid methodology for holistic testing.

tructure[1], so the question "what infrastructure is available to carry out the test?" is addressed to define the available RIs' capabilities in step ②. Finally, by answering "how should the available infrastructure be configured to carry out the specified test in a concrete experiment" in step ④, the experiment specification is completed.

In order to formulate the holistic test case, three inputs are required:

- A *Generic System Configuration*, which is a description of the overall system configuration (and assumptions) within which we seek to test an object (or subsystem).
- A set of *Use Cases* that describe the sequence of actions/functions that are expected of the tested object.
- The *Test Objective*, which is the purpose for carrying out the tests, stating the overall evaluation objective.

Based on the previous inputs, the **test case** describes the following concepts:

- A *System under Test* (SuT) that identifies the system boundaries of an abstract test system entailing all relevant interactions requiring investigation.
- The *Object(s) under Investigation* (OuI) identifying the systems, subsystems, or components within the SuT to which the test criteria will be applied.
- The *Domain(s) under Investigation* (DuI), which identifies the relevant physical or cyber-domains (and sub-domains) which are of interest for the test parameters and connectivity.
- With respect to the use cases, *Function(s) under Test* (FuT) and *Function(s) under Investigation* (FuI), which describe the relevant operations relevant to the SuT (in the FuT) and the ones that are being investigated with respect to the OuI (in the FuI).

---

[1]The terms lab, research infrastructure (RI), test infrastructure, as well specific co-simulation environments are used interchangeably here.

TABLE I
TYPES OF SYSTEM CONFIGURATIONS

Name / Purpose	Context	Generic/Specific	SCType	Explanation
Function-System Alignment	Use Case	GSC	UC-GSC	As SGAM domains & zones: reference designation for functions, independent of test case.
Test Case context model	Test Case (step ①)	GSC	TC-GSC	Establishes type conventions for test case: relevant SC component types, domains, etc., and categorically identifies the SuT.
Test System	Test Specification (step ③)	(S)SC	TS-SC	A concrete instance of TC-GSC to address a specific OuI and specific set of test criteria.
Experiment Setup	Experiment Specification (step ④)	(S)SC	E-SC	The configuration and interconnection of RI components, representing the SuT, and including OuI.
RI description	RI database (step ②)	(S)SC	RI-SC	Configuration and components of an RI, including *potential* multiplicity and connectivity of RI components.
RI information model	RI profiling	GSC	RI-GSC	RI profiling information model, here specific to CPES laboratories.

- The *Purpose of Investigation* (PoI), which details the test objective and sub-objectives by qualifying each objective as either *validation*, *verification*, or *characterization*.
- *Test Criteria*, which state the metrics that need to be evaluated for each of the objectives formulated in the PoI.

With these concepts, we are able to define a **test specification**, which consists of:

- A *test system configuration* which defines how the OuI is going to be embedded in the SuT.
- The *input/output* parameters of the system which will be varied, observed, and evaluated.
- The *test design*, which defines the manner in which the test will be carried out.

Finally, given a test specification, we can map the testing requirements with the capabilities of one or more RI(s) in order to carry out the experiment. This mapping (step ④ in Fig. 1) leads to a concrete **experiment specification**, which identifies the laboratory components and devices that reflect the required functions, domains and connections specified in the previous steps in order to execute the test.

The experiment specification is the last step of the description phase before an experiment can be conducted (step ⑤ in Fig. 1). After the experiment, a pre-assessment of the results is required (step ⑥) to decide whether the test specification will have to be adapted for a re-run of the experiment in a modified setting, or if the results are suitable for a final evaluation of the experiment (test evaluation, step ⑦).

The process outlined above is equally applicable to tests involving multiple RIs. Examples of multi-RI testing include the mapping of identical test specifications to different RIs or the synchronous execution of one large experiment spanning multiple communicating RIs. However, each of these cases introduces specific additional requirements such as ensuring the comparability of results obtained on dissimilar laboratory hardware, or the need to specify the real-time interaction at the inter-RI level. These needs are being addressed by the ERIGrid project but are beyond the scope of this paper.

B. System Configurations (SC)

The specification of system configurations is central to the test description, similar to other related specification work (e.g., smart grid use cases with reference to the Smart Grid Architecture Model (SGAM), information modeling for power system ICT via the Common Information Model (CIM), or other applications of the Unified Modeling Language (UML) or the Systems Modeling Language (SysML)). To integrate the SC description methods used in smart grid disciplines (e.g., electrical, ICT, or thermal systems), ERIGrid adopted and generalized the basic system description concepts that are employed in the power systems CI). To avoid overly complex specification details, the following upper ontology model has been identified: *System*s are composed of *Components* and are themselves components. Components have *Terminals*, which may have a directionality and are associated with one *Domain*. Domains can be structured hierarchically. Two or more terminals associated with the same domain can be connected using a *Connection Point*. All of the above are *System Configuration Objects*, and a set of them composes a *System Configuration Container*, which has a system configuration type (*SCType*) attribute. *Constraints* can be associated with any type of system configuration object. Table I lists the six relevant types of system configurations.

A domain-independent graphical realization of an SC description (TS-SC) is found in Fig. 3, where a test system is specified. Using another graphical convention, Fig. 5, defines a co-simulation Experiment SC (E-SC). When formulating the E-SC based on TS-SC, interactions among SuT elements are fully represented, whereas non-SuT components (non-grayed background in Fig. 3), are represented by simple equivalents.

While the SC abstraction is useful, the number of SC variants listed in Table I may be confusing. Two main aspects of distinction should be noted: Firstly, there are *RI-oriented system configurations* (RI-GSC, RI-SC,E-SC) and *real-world oriented system configurations* (UC-GSC, TC-GSC, TS-SC). The former describe the RI capabilities and concrete RI objects that are eventually present in an experiment. This may include a physical grid or amplifier in a lab, but also simulation facilities. The latter types of SC aim to represent aspects of the real-world. Secondly, there are *generic* and *specific* system configurations. The generic SCs (UC-GSC, TC-GSC,RI-GSC) define types or classes of objects, whereas the specific SCs (TS-SC, RI-SC, E-SC) define concrete instances of SC elements.

III. Co-simulation of CPES using Mosaik and FMI

The software co-simulation in ERIGrid is realized by employing the framework mosaik as a co-simulation master and interfacing all simulators via the Functional Mock-up Interface (FMI) standard. In the following, the components of this co-simulation platform are briefly introduced. Subsequently, the coupling between mosaik and FMI is elucidated, as well as the utilization of the platform in the project.

A. Framework and Interfaces

1) Mosaik: Mosaik is a co-simulation framework that is designed for easy integration of simulators and flexible creation of co-simulation scenarios. These design goals are achieved via two Application Programming Interfaces (API). The Component-API allows integration of simulators via description of the accessible data and a set of interface functions. The API has been ported to different programming languages to provide a broad user support. The Scenario-API provides a set of functions to establish data flow connections between simulators and execute the co-simulation. The execution is managed by a discretely timed scheduling algorithm. Two versions of mosaik have been developed so far. Mosaik 1 [16] incorporates a domain specific language for the two API types with the goal of automatic consistency checking of co-simulation scenarios. It ultimately proved to be too inflexible for practical work. Mosaik 2 (e.g. [17], [18]) is an enhanced, streamlined version of the software that is purely based on Python and provides more concise and flexible APIs.

2) Functional Mock-up Interface: FMI [19] is a standard for the interfacing of simulators and simulation models that is supported by various modeling and co-simulation tools. It is comparable with the mosaik Component-API in the sense that it provides a (XML-based) data model and a set of interface functions. However, FMI is more complex and powerful than its mosaik counterpart. The reason for this lies in the fact that the Component-API is specifically geared to the mosaik scheduling algorithm while FMI has been designed for interaction with a variety of master algorithms. The design of the master is not part of the FMI standard so that the complexity of FMI is the price of its flexibility. A simulator that employs the FMI standard is capsuled in a Functional Mock-up Unit (FMU). There are two types of FMI so that FMUs may possess different structures. FMI for Co-Simulation (FMI-CS) assumes that the FMU includes a solver and thus may independently simulate when called. FMI for Model Exchange (FMI-ME), on the other hand, expects the master algorithm to solve the simulation model provided by the FMU.

To facilitate the handling of FMUs, the FMI++ toolbox has been developed [20]. It provides a more high-level set of interface functions that is still usable by a variety of master algorithms. Furthermore, it includes a number of utility methods, e.g., for rollback and interpolation, as well as a set of readily usable integrators to solve FMI-ME systems.

B. Coupling between mosaik and FMI

Mosaik is used as the co-simulation framework in ERIGrid due its good usability while providing robust scheduling at the same time. For the simulator interfacing, however, FMI is employed due to its wide acceptance and usefulness for potential future exchange of schedulers. Therefore, a mapping between the Component-API and FMI has to be established to grant mosaik access to simulators encapsulated in FMUs. Such a mapping has already been conducted successfully by [21] via the use of the FMI++ toolbox. However, their approach employs mosaik 1 and an outdated version of FMI++. Furthermore, only FMI-ME is supported whereas ERIGrid will utilize both types of FMI. As a consequence, coupling is suggested that employ the more streamlined interface functions of mosaik 2 and the new FMI++ version. Separate interfaces are established for FMI-ME and FMI-CS. Each of them is kept as generic as possible so that integration of several FMUs of the same type requires only an adjustment of parameter sets.

As shown in [21], a mapping has to be established between the mosaik API functions and the FMI++ functions. ERIGrid employs a mapping similar to the one presented in that work, although using updated function references. Similarly, the XML-based description of model variables has to be mapped onto the mosaik model description. FMI includes a complex variable description with data fields like "variability" and "causality", many of which do not possess a counterpart in mosaik. Parameter-type variables are also parameters in mosaik. Variables with the causality "input" or "output" are summarized as "attributes". Nevertheless, the exact causalities as well as the data types (real, integer, etc.) have to be stored within the FMI-mosaik interface in order to select the correct getter and setter functions provided by FMI++.

As mentioned before, FMI-ME expects the master algorithm to solve the simulation model. This notion generally conflicts with the co-simulation concept, e.g., mosaik does not include any solvers. This problem is circumvented by employing the integrators provided by FMI++ within the FMI-mosaik interface. Thus, whenever mosaik calls for the FMU to execute a step, the interface solves the model for this time step. The desired integrator type can be set, along with other technical details like the time step size, as a "simulator parameter". FMI-ME is, in summary, supported via a form of "co-simulation via capsuling" as illustrated in Fig. 2.

C. Application in ERIGrid

Mosaik fullfills two roles in ERIGrid. On the one hand, it acts as an environment for exemplary co-simulation test cases, serving its usual purpose. This setup utilizes the coupling between mosaik and FMI-CS to allow integration of complex tools like PowerFactory. On the other hand, mosaik is also used as a testbed for a library of newly developed smart grid component models. These models are supplied following the FMI-ME standard to enable versatile future use.

The developed models should be capable of being integrated into a variety of different simulation environments like MATLAB or OpenModelica. Therefore, FMI-ME is the
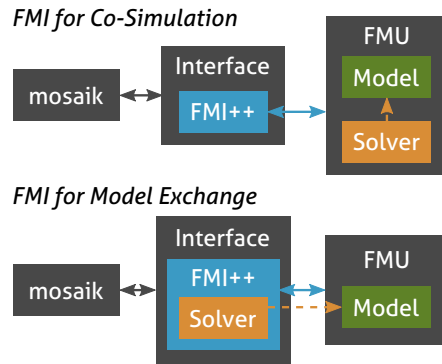
Fig. 2. Interfacing of mosaik for the different FMI types.

preferred standard since it provides more freedom in the model usage. Due to the interface presented above, mosaik is still able to integrate these FMUs and serve as an environment for integration testing. For testing, an FMU is integrated and presented with pre-defined input data from a generic data source component. The output may be stored in a database component and analyzed for unexpected behavior. As a matter of fact, mosaik's flexible Scenario-API allows a convenient way to automatically test sensitivities towards different parameter values, temporal resolutions, integrator types, and so forth. Mosaik's capabilities for such a type of black box analysis have already been demonstrated in [18].

IV. SOFTWARE AND COMPONENT INTERFACES APPLIED IN ERIGRID

Challenges in transferring R&D results into real-world applications often arise already in the design phase, due to incompatible simulation-based approaches adopted by researchers and industry that prevent a consolidated solution for validation. Hence, one goal of the ERIGrid project is the demonstration of the feasibility of FMI-based co-simulation and model exchange for validating Smart Grid applications, in order to promote tool and model interoperability and stress the importance of this topic in the FMI community. To this end, the proof-of-concept studies performed in ERIGrid need to be representative and demonstrate the added value of such an approach. In the following, the rationale behind the choice of tools and models in this context is explained.

A. Simulation domains

Modeling and simulation of smart grid scenarios comprise aspects from various technical domains, with a broad variety of available simulation tools for each of these domains. For the purpose of ERIGrid, three particular domains have been identified:

1) *Power systems*: The technical infrastructure for the distribution of electricity is obviously the core element for smart grid applications. For ERIGrid, POWERFACTORY[2] will be primarily used to simulate power systems.

2) *Communication*: Virtually any smart grid application relies in one way or the other on information exchange. The discrete-event network simulator NS-3[3] has been selected as main tool for ERIGrid to simulate dedicated and general-purpose ICT systems.

3) *Automation and control*: The potential complexity of controllers used in smart grid applications necessitates dedicated tools for their implementation. For ERIGrid, MATLAB/SIMULINK[4] will be primarily used to provide the needed controller functionality for simulations.

Furthermore, a dedicated library of models will be developed using OPENMODELICA and MATLAB/SIMULINK, which will complement and expand the functionality provided by the domain-specific tools. The selection of the tools and models was driven by several factors, such as previous experience of partners, availability of FMI-compliant interfaces or details regarding licenses.

B. Tool compatibility

Each of the selected tools represents the state-of-the-art in their respective domain. As such, this selection provides a representative case that can serve as a relevant proof-of-concept for smart grid co-simulation. However, the selected tools implement very contrasting modeling and simulation paradigms and to prove the capability and point out the shortcomings of FMI-compliant interfaces to successfully handle this heterogeneous combination is a key issue.

Power system modeling: Tools for this domain rely on continuous time-based modeling paradigms, typically representing individual components by (sets of) differential algebraic equations. They enable the simulation of the evolution of the system state either with the help of models that depend explicitly on time (RMS and EMT simulation) or by computing a series of subsequent power flow calculations (quasi-static simulation). Even though POWERFACTORY does not offer the functionality to export models in a way that is compliant with FMI-ME, it provides an API that allows to interact with it [22]. The functionality of this API has been successfully mapped to the that specified for FMI-CS[5].

Communication network modeling: Simulators for this domain use abstractions of the deployed hardware and software that allows the representation of communication processes as a sequential processing and transmission of (virtual) messages and signals. Hence, communication network simulators commonly implement discrete event-based simulation paradigms, where each event marks a significant step of message processing or transmission. The network simulator NS-3 has already been successfully used for co-simulation, see for instance [23], but no FMI-compliant interface is available so far. Based on previous theoretical work dealing with the capability of FMI to handle discrete event semantics [24], [25], an interface according to the FMI-CS specification is being developed in ERIGrid for NS-3.

---

[2]See http://www.digsilent.com/

[3]See https://www.nsnam.org/
[4]See http://www.mathworks.com
[5]See http://powerfactory-fmu.sourceforge.net

Automation and control: For implementing control algorithms MATLAB/SIMULINK has been chosen, due to its versatility as well as its popularity and widespread use for this purpose. The capability of FMI to encapsulate the functionality of discrete event-driven controller models has been already discussed in [26]. SIMULINK implements a continuous time-driven paradigm and allows to translate models into FMUs for ME (discussed next in Section IV-C). MATLAB on the other hand is a multi-purpose computational environment, which adheres to no specific modeling paradigm and provides by itself no notion of time. However, it is possible to put MATLAB's full functionality at the disposal of the user via an approach that is compatible with FMI-CS[6].

C. Model library

The model library developed for ERIGrid aims to facilitate and accelerate the integration and validation of smart grid solutions by extending the functionality of available tools with specific models. To this end, the models will be compliant to the FMI-ME specification, in order to ensure a tool-independent implementation that improves reusability. However, even though FMI-ME is a well accepted specification, basically no present-day proprietary power systems simulation tool has incorporated the FMI-ME standard so far. By demonstrating its use and potential benefits within ERIGrid, it is hoped that the adoption of FMI-ME within the power industry is further propelled.

In the following, examples of models being developed for each of the considered simulation domains and their potential use to the wider community are presented. All the models will be developed either in MATLAB/SIMULINK or OPENMODELICA, which allow to export models as FMUs for ME.

Power system components: A number of models specific to power systems are currently under development for FMI-ME, including a reduced-order distribution dynamic equivalent model, FlexHouse thermal model, PV and battery models, and aggregated wind turbine dynamic model. This set of tool-independent models can potentially be used for different simulation studies and proof of concepts.

For example, the reduced-order dynamic equivalent model represents an actual $115\,\text{kVA}$ smart grid laboratory at the University of Strathclyde. Although simulation tools such as POWERFACTORY provide many component models, this reduced-order equivalent model will allow for the dynamics of a real smart grid laboratory to be incorporated in simulation studies. This enables large-scale system studies to be undertaken with the least amount of computational requirements.

Communication network modeling: The difficulty of incorporating the effects of communication within a power system simulation tool (due to its discrete event-based nature) often leads to simplistic fixed-time delay models being utilized. However, within ERIGrid, a representative communications model that can be easily utilized within a power system simulation is under development. It will rely on various communication network parameters as inputs and will calculate

[6]See http://matlab-fmu.sourceforge.net

delays based on stochastic equations taking into consideration a set of defined uncertainties [27]. By employing the IEC 61850 GOOSE protocol, the model will delay a signal based on a value chosen by means of a Gaussian distribution. This will enable power system studies to incorporate a simple yet more realistic, representative delay.

Automation and control: A large number of smart grid applications increasingly depend upon reliable measurements being obtained from within the network. Recently, network critical applications, such as protection, are dependent upon utilizing accurate measurements from a large number of *Phasor Measurement Units* (PMU). Within ERIGrid, both P-Class and M-Class PMU models compliant with IEEE C37.118.1a are being developed as in [28]. These models can be utilized by the wider community for development of novel applications that rely on data from PMUs.

V. HOLISTIC TESTING: PROOF-OF-CONCEPT BY CO-SIMULATION

A. Modelling and Simulation needs

Generally speaking, software experiments can be subdivided into monolithic software experiments (one domain, one tool), multi-domain tools (multiple domains, one tool), hybrid models (multiple tools, one domain), and heterogeneous modeling (i.e., multiple domains and tools). The latter is predominantly realized with co-simulation and will be the main focus of the proof-of-concept of the above proposed holistic test case formalization [8], [10].

The application of the mosaik framework and the FMI opens up a massive spectrum of co-simulation possibilities, but gives also rise to development and implementation challenges. It has been decided to define three exemplary test cases that at least contain one co-simulation experiment, each of which treats one specific development need:

1) interfacing software tools with physical controllers
2) signal-based synchronization
3) cyclic dependencies between simulation federates

Software/hardware interfacing will be studied using an on-load tap changer and its control. Both can be implemented as hardware or emulated while synchronizing in wall clock time. Signal-based synchronization is addressed by an on-load tap changer that is regulated by a distributed voltage controller, the ICT aspects of which play a prominent part in the overall behaviour of the SuT.

In the forthcoming we will focus on the so-called cyclic dependency issue that arises within mosaik. Coupling continuous simulators necessitates synchronization of interface variables during all stages of execution. By default, however, mosaik only allows the specification of one-way dependencies between simulation federates and defining cyclic dependencies is non-trivial. To test this particular feature, a rather simple simulation setup that still allows studying the problem would be optimal. For the sake of realism we use a power system model, being simulated in tool 1, and an aggregated wind power plant model, which is simulated in tool 2. This test

bed potentially allows the inclusion of controls and phenomena with diverse time constants (e.g., supervisory frequency control, voltage control, fault-ride through) and of different nature (i.e., discrete versus continuous).

B. Test Case (step ① in Fig. 1)

1) Use Case and Function under Test: The Use Case treated here is the capability of a Wind Power Plant (WPP) to stay connected during a three phase short circuit inside the transmission system. This capability is referred to as Fault Ride-Through (FRT) and is often laid down in grid codes, which require conformance at the coupling point (i.e., the legal boundary between the power park module owner and the transmission system operator)[29]. Hence, all individual wind turbines need to jointly fulfill this common requirement. Aside from FRT the WPP needs to comply to a myriad of other regulations. Essential for FRT because of the mutual interaction are reactive power and voltage control. Thus, the FuT are 1) FRT capability and 2) reactive power control. The PoI is to *validate* the compliance of the WPP as a whole to the voltage against time profile during voltage sags and against a voltage-reactive power curve for normal operating conditions.

2) System Configuration and SuT: The test system SC (see Table I for the terminology) and the SuT of this test case are shown in Fig. 3. It consists of the transmission system, the collection grid of the WPP, the individual Wind Turbine Generator (WTG) and its controls. The connections between the components and subsystems reside in a domain, in this case electric, control, or environment.

The SuT is the part of the system configuration comprising the FuT. Although the actual implementations of FRT compliance are done inside the individual wind turbine controllers, FRT verification is required at the coupling point. Hence the boundary of the SuT is between the transmission system and the WPP collection grid. By means of vector control the protection circuits in the DC-link dynamically separate the electromechanic part of the WTG from the grid interface. This allows us to not take the wind turbine itself (i.e., aerodynamic conversion, drive train, electromechanic conversion, stator-side controls, pitching, etc.) into consideration. This also bounds the DuI to the electric and control/ICT domains.

As for verification of voltage and reactive power control the interactions are foreseen to mainly occur in the collector grid. A supervisory WPP controller usually sets reference values for either voltage, reactive power, or power factor, and the individual WTGs need to track these set points locally. Interaction with FRT is foreseen in case additional reactive current injection is engaged during faults. The SuT for this FuT hence comprises the WPP collection grid, the grid-side converter, and its vector controls.

3) Test Criteria and FuI: The set of system configuration, use cases, and functions under test cater for the evaluation of two specific test criteria, namely

1) The validation of the ability of the WPP to maintain synchronism with the external system during voltage dips, as defined in a voltage against time curve;
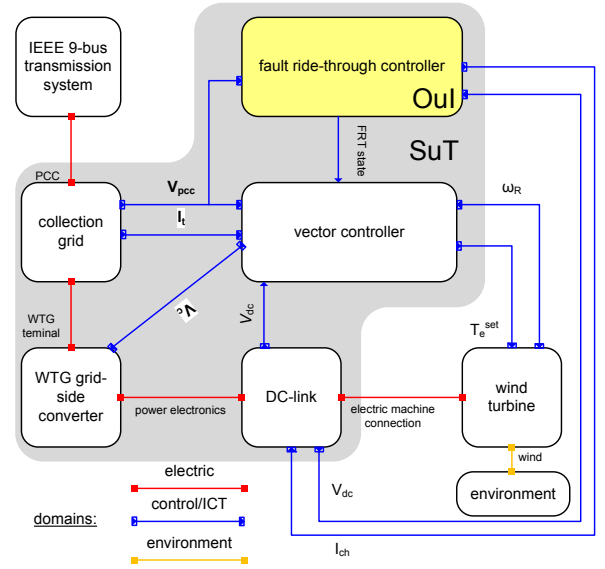

Fig. 3. Test system configuration (TS-SC) of the example holistic test case.

2) The validation of the conformance of the WPP to track a reactive power against voltage curve by means of a centralised voltage control mechanism.

To test the former, both functions under test are significant as reactive power control during faults may boost the voltage magnitude, alleviating the FRT duty of the WPP as a whole. Hence, the FuI are the same as the FuT for the first test criterion. For the second, the FRT mechanism is superfluous to consider as the criterion is merely valid for normal operating conditions. Thus, the FuI is the centralised voltage control of the WPP.

C. Test Specification (step ③ in Fig. 1)

For the sake of brevity, we limit the test specification to the FRT compliance and assume a similar procedure for evaluating the test criterion concerning the reactive power at the coupling point. The FuI here is the set of actions the FRT controller has to take to fulfill the test criterion. Hence, the OuI is the FRT controller. At this stage we need to further quantify the test criterion, and need to define a more specific system configuration.

1) Specific Test Criterion and Test System Configuration: The test criterion is refined according to Fig. 4. It shows the time versus positive-sequence voltage profile that the WPP needs to be able to endure. In case the point of common coupling (PCC) voltage enters the grey area (i.e., $U^{\mathrm{PCC}} < U_{\mathrm{ret}}$, Ⓑ) the WPP is allowed to disconnect from the grid. In zone Ⓐ, the FRT controller must engage internal component protection and while is shall prevent entering zone Ⓑ. The protection and additional control measures that need to be taken are implemented as a finite state machine, hence a discrete controller [30]. The corresponding test system comprises the IEEE 9-bus transmission system, the SuT and the WTG. The SuT and its internal connections are indicated in Fig. 3 by the gray area. The IEEE 9-bus benchmark system (i.e.,
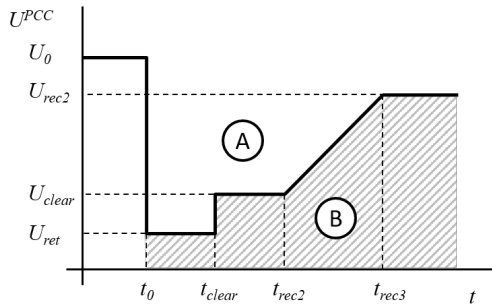
Fig. 4. Fault ride through voltage profile.

Fig. 5. Experimental implementation of TC1 in terms of mosaik and FMI.

[31]), has been adapted to $50\,\mathrm{Hz}$ and to contain more realistic dynamic behaviour of loads. This TS-SC represents the WPP aggregatedly by a single permanent-magnet, direct drive WTG. Detailed single-line representations of the collection grid and transmission grid have been omitted for brevity.

2) Overall design of the test: The test criterion to verify consists of two parts, a deep-dip part between t_0 and t_{clear} and a recovery part between t_{clear} and t_{rec3}. The deep-dip part is commonly caused by severe voltage dips that are quickly isolated by protection. The recovery part serves two situations: 1) the voltage amplitude dynamics after fault clearance at or before t_{clear}, 2) the voltage response after a shallow voltage dip cleared after t_{clear} and does not violate $U^{\mathrm{PCC}} < U_{\mathrm{clear}}$.

These situations can be replicated by causing a voltage dip at the PCC by means of a self-extinguishing 3-phase-to-ground fault in the external grid. The testing procedure is as follows:

1) Determine operating point;
2) Set short circuit location to x in the IEEE 9-bus system, corresponding dip depth; U_{ret}^x
3) Initiate short circuit at $t_0 = 0.1\mathrm{s}$;
4) Clear fault at $t_{\mathrm{clear}} = y$;
5) Assess test criteria; and
6) Vary x and y and repeat above sequence 2-5 times.

The experiment is considered successful if the WPP maintains synchronism under the circumstances defined above.

D. Experiment Specification (step ③ in Fig. 1)

The test criterion can be evaluated by several experiments. As field testing or pure hardware testing is unfeasible in this case—it will be challenging to convince DSOs and WPP owners to release their system as a testbed—the emphasis will be on pure software and (C)HIL experiments. Example experiment realizations include:

- Monolithic simulation using MATLAB/Simulink (Simscape Power Systems Toolbox)
- CHIL in which the grid is simulated by a real time simulator and one of the wind turbine controllers is implemented into a PLC and is connected via I/O
- Co-simulation between PowerFactory and OpenModelica
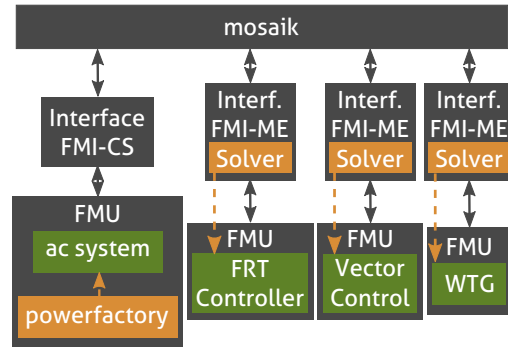- Co-simulation between PowerFactory and mosaik using both FMI for CS and ME

The latter will now be explained in more detail as it combines all the previously introduced building blocks (i.e., FMI++, mosaik, simulation tools). Fig. 5 shows the distribution of the system configuration across POWERFACTORY and FMUs for ME. The ac grid part of the test system is simulated in POWERFACTORY and comprises the IEEE 9-bus system, the collection grid, and the ac grid interface of the WTG. The latter is modelled by a variable quasi-stationary current source. The set point values of this current source typically come from the master simulator, which is done through the FMI-CS interface. This wrapper connects to mosaik, which on its turn orchestrates the overall simulation procedure. The wind turbine, its controls, and the FRT controller are tested and implemented in SIMULINK and subsequently exported as FMUs for ME. The FMI++ wrapper accounts for the solution routine needed for these FMUs.

This experimental setup combines all the merits that come along with mosaik and FMI++: it allows design and validation of models in a multi-domain simulation environment, couple it to a master simulator by FMI-ME, and a flexible mock-up with a commercial, domain-specific simulation tool by FMI-CS.

VI. EVALUATION, DISCUSSION, AND OUTLOOK

This paper discussed a formal approach for holistic test description, which is an invaluable tool for evaluation of the integrated cyber-physical systems. This approach aims to harmonize the concepts and testing methods used by laboratories and research infrastructures to analyze CPES, which are a combination of different technologies across heterogeneous domains (power, ICT/ automation, markets, etc.). In this paper we provided co-simulation by the open-source mosaik framework using the FMI standard as one particular experimental implementation of the holistic test description approach. The extension of mosaik with FMI for co-simulation and model exchange offers a versatile tool chain that allows multi-domain co-simulation, testing, and validation, for components and systems alike.

As a proof of concept of holistic CPES assessment by co-simulation, this paper discussed the fault ride-through of a wind power plant that is grid connected at sub-transmission level. This test case highlights the benefits of the formal specification method. On use case level it clearly distinguishes

the interacting functions under test, on test specification level the behaviour and modeling assumptions can be projected on the test criteria, and the experimental specification allows a very flexible way of assessing the separate sub-systems on diverse experiment platforms. Next steps include the test evaluation criteria, test refinement, and optimally mapping the research infrastructure to the experiment specification. Future work will include the application of uncertainty quantification (UQ) methods to the test specification and evaluation process, including UQ annotations in the model library, which will enable UQ of the associated co-simulation experiments [32].

The tools and models chosen and implemented in ERIGrid provide a representative example of FMI-compliant state-of-the-art concepts for modeling and simulation of CPES. This will pave the way for improving interoperability and repeatability for simulation-based evaluation and validation approaches, notably by demonstrating the feasibility and illustrating the advantages and shortcomings of using FMI-compliant tools and models.

REFERENCES

[1] European Commission, "Energy roadmap 2050," Smart Grid Task Force, Brussels, Communication from the commission to the European parliament, the council, the European economic and social committee and the committee of the regions, 2011.

[2] P. Palensky and F. Kupzog, "Smart grids," *Annual Reviews of Environment and Resources*, vol. 38, pp. 201–226, 2013.

[3] T. Strasser, F. Andrén, J. Kathan, C. Cecati et al., "A review of architectures and concepts for intelligence in future electric energy systems," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 4, pp. 2424–2438, 2015.

[4] H. Farhangi, "The path of the smart grid," *IEEE Power and Energy Magazine*, vol. 8, no. 1, pp. 18–28, 2010.

[5] T. Strasser, F. Pröstl Andrén, G. Lauss, et al., "Towards holistic power distribution system validation and testing—an overview and discussion of different possibilities," *e & i Elektrotechnik und Informationstechnik*, vol. 134, no. 1, pp. 71–77, 2017.

[6] R. Podmore and M. Robinson, "The role of simulators for smart grid development," *IEEE Transactions on Smart Grid*, vol. 1, no. 2, pp. 205–212, Sept. 2010.

[7] K. Mets, J. A. Ojea, and C. Develder, "Combining power and communication network simulation for cost-effective smart grid analysis," *IEEE Communications Surveys Tutorials*, vol. 16, no. 3, pp. 1771–1796, 2014.

[8] P. Palensky, A. A. van der Meer, C. D. López, A. Jozeph, and K. Pan, "Co-simulation of intelligent power systems - fundamentals, software architecture, numerics, and coupling," *IEEE Industrial Electronics Magazine*, vol. 11, no. 1, Mar. 2017.

[9] "European Research Infrastructure supporting Smart Grid Systems Technology Development, Validation and Roll Out (ERIGrid)," https://www.erigrid.eu, accessed: 2017-02-19.

[10] P. Palensky, A. A. van der Meer, C. D. López, A. Jozeph, and K. Pan, "Applied co-simulation of intelligent power systems: implementation, usage, examples," *IEEE Industrial Electronics Magazine*, vol. 11, no. 2, Jun. 2017.

[11] R. Santodomingo, M. Uslar, A. Goring, M. Gottschalk, L. Nordstrom, A. Saleem, and M. Chenine, "SGAM-based methodology to analyse smart grid solutions in discern european research project," in *2014 IEEE International Energy Conference (ENERGYCON)*, 2014, pp. 751–758.

[12] K. Heussen, M. Uslar, and C. Tornelli, "A use case methodology to handle conflicting controller requirements for future power systems," in *2015 International Symposium on Smart Electric Distribution Systems and Technologies (EDST)*, 2015, pp. 582–587.

[13] CEN-CENELEC-ETSI, Smart Grid Coordination Group, "Smart Grid Reference Architecture (SGAM), v 3.0," 2012.

[14] ——, "SG-CG Report on Interoperability," 2014.

[15] M. Blank, S. Lehnhoff, K. Heussen, D. E. M. Bondy, C. Moyo, and T. Strasser, "Towards a foundation for holistic power system validation and testing," in *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2016, pp. 1–4.

[16] S. Schütte, "Simulation Model Composition for the Large-Scale Analysis of Smart Grid Control Mechanisms," PhD, Carl von Ossietzky University of Oldenburg, 2013. [Online]. Available: http://oops.uni-oldenburg.de/1768/

[17] M. Büscher, A. Claassen, M. Kube, S. Lehnhoff, K. Piech, S. Rohjans, S. Scherfke, C. Steinbrink, and J. Velasquez, "Integrated Smart Grid Simulations for Generic Automation Architectures with RT-LAB and mosaik," in *IEEE International Conference on Smart Grid Communications (SmartGridCom)*, 2014.

[18] S. Lehnhoff, O. Nannen, S. Rohjans, F. Schlogl, S. Dalhues, L. Robitzky, U. Häger, and C. Rehtanz, "Exchangeability of power flow simulators in smart grid co-simulations with mosaik," in *2015 Workshop on Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES)*, 2015.

[19] MODELISAR Consortium, "Functional Mock-up Interface for Model Exchange and Co-Simulation," Tech. Rep., 2014.

[20] E. Widl, W. Müller, A. Elsheikh, M. Hörtenhuber, and P. Palensky, "The FMI++ library: A high-level utility package for FMI for model exchange," in *2013 Workshop on Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES)*, May 2013.

[21] S. Rohjans, E. Widl, W. Müller, S. Schütte, and S. Lehnhoff, "Gekoppelte simulation komplexer energiesysteme mittels mosaik und FMI," *at – Automatisierungstechnik*, vol. 62, no. 5, pp. 325–336, 2014.

[22] M. Stifter, R. Schwalbe, F. Andrén, and T. Strasser, "Steady-state co-simulation with PowerFactory," in *2013 Workshop on Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES)*, May 2013.

[23] S. Ciraci, J. Daily, J. Fuller, A. Fisher, L. Marinovici, and K. Agarwal, "FNCS: A framework for power system and communication networks co-simulation," in *Proceedings of the Symposium on Theory of Modeling & Simulation - DEVS Integrative*, ser. DEVS '14. San Diego, CA, USA: Society for Computer Simulation International, 2014, pp. 36:1–36:8.

[24] D. Broman, C. Brooks, L. Greenberg, E. A. Lee, M. Masin, S. Tripakis, and M. Wetter, "Determinate composition of FMUs for co-simulation," in *13th International Conference on Embedded Software (EMSOFT)*, September 2013.

[25] S. Tripakis and D. Broman, "Bridging the semantic gap between heterogeneous modeling formalisms and FMI," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2014-30, Apr 2014.

[26] E. Widl, F. Judex, K. Eder, and P. Palensky, "Fmi-based co-simulation of hybrid closed-loop control system models," in *2015 International Conference on Complex Systems Engineering (ICCSE)*, Nov 2015.

[27] P. Dambrauskas, M. H. Syed, S. M. Blair, J. M. Irvine, I. F. Abdulhadi, D. E. M. Bondy, and G. M. Burt, "Impact of realistic communications for fast-acting demand side management," in *proc. CIRED Workshop 2017*, Glasgow, UK, Jun., "12–15", 2017.

[28] A. J. Roscoe, I. F. Abdulhadi, and G. M. Burt, "P and m class phasor measurement unit algorithms using adaptive cascaded filters," *IEEE Transactions on Power Delivery*, vol. 28, no. 3, pp. 1447–1459, Jul. 2013.

[29] European Commission, "Commission regulation (EU) 2016/631," 2016, last accessed: Mar. 2017. [Online]. Available: http://eur-lex.europa.eu/eli/reg/2016/631/oj

[30] A. A. van der Meer, M. Ndreko, M. Gibescu, and M. A. M. M. van der Meijden, "The effect of FRT behavior of VSC-HVDC connected offshore wind power plants on AC/DC system dynamics," *IEEE Transactions on Power Delivery*, vol. 31, no. 2, pp. 878–887, Apr. 2016.

[31] P. M. Anderson and A. A. Fouad, *Power System Control and Stability*. Wiley-IEEE Press, 2003.

[32] C. Steinbrink, S. Lehnhoff, and T. Klingenberg, "Ensemble-Based Uncertainty Quantification for Smart Grid Co-Simulation," in *Measurement, Modelling and Evaluation of Dependable Computer and Communication Systems*, 2016.