Requirements for Real-Time Hardware Integration into Cyber-Physical Energy System Simulation

Mario Faschang, Friederich Kupzog, and Edmund Widl
AIT – Austrian Institute of Technology
Vienna, Austria
Email: [mario.faschang; friederich.kupzog; edmund.widl]@ait.ac.at

Sebastian Rohjans and Sebastian Lehnhoff
OFFIS – Institute for Information Technology
Oldenburg, Germany
Email: [rohjans; lehnhoff]@offis.de

***Abstract*—Modeling and simulation are essential for the development and assessment of new technologies for complex cyber-physical systems, both in academia and industry. One particular instance of such complex systems are novel energy systems. They comprise a variety of heterogeneous physical domains (electricity, thermal, gas, etc.), modeled as time continuous dynamic systems, and time discrete communication and control systems. Tools and solutions based on co-simulation concepts for such cyber-physical energy systems (CPES) have been developed in recent years. This paper focuses on the need for real-time hardware integration into CPES simulation, identified from recently conducted research projects. From those projects – presented as use cases – relevant actors are identified and their interaction and data exchange is discussed. Resulting requirements are presented for the CPES simulation and the hardware to be integrated. The handling of heterogeneous laboratory hardware is discussed and a simplification in the form of a hardware abstraction layer is proposed.**

I. Introduction and current state

The step-wise evolution of classical power grids towards reliable and sustainable future power grids – better known as *smart grid* – creates a set of new opportunities and challenges. The introduction of information and communication technology (ICT) in the power grid infrastructure along with novel components, e.g., remotely controllable distributed generation units, results in a complex and distributed system of systems. For the design and operation of such cyber-physical energy systems (CPES), measures for the prediction of its behavior and for testing of its components are inevitable. Approaches utilizing co-simulation (co-operative simulation) concepts have proven to be suitable for analyzing the impact of new smart grid technologies [1]–[4], as they allow for a detailed and large scale assessment of such systems. Although the term co-simulation is not uniquely defined, it describes in general the joint simulation of models developed with different tools between which intermediate results are exchanged during simulation execution. However, between their discrete communication points the subsystems are solved independently [5].

In the context of CPES, the implementation focus of co-simulation setups can be either software-centric [6]–[9], hardware-centric (hardware-in-the-loop, HIL) [10] or follow a hybrid hardware/software concept [11]–[13]. Experience from these different approaches shows that hybrid concepts, which integrate extensive simulation capabilities with real-world lab set-ups, allow for more comprehensive tests of novel smart grid components, especially before a roll-out to the field. Following the requirements for future smart grid simulation tools outlined in [14], the key for developing future smart grid simulation tools is to not "re-invent the wheel", but rather to re-use already established models and simulators. As such, the authors of this work aim to develop a flexible solution for hardware integration into CPES simulations, for practical smart grid applications with a focus on lab evaluation and certification of related components. Within this paper the goal is system analysis and requirements extraction for the hardware integration into existing CPES simulation frameworks.

The remainder of this paper is structured as follows. Section II explains the context for this work, focusing on use cases identified in recent research activities. These use cases are then analyzed to identify a comprehensive generalized use case. Based on this, the relevant actors and their interactions are illustrated, together with the different types of data to be exchanged among these actors. Section III discusses the generalized use case in more detail, covering real-time hardware integration into co-simulation approaches for CPES in depth. From this description the concrete requirements are derived and listed in Section IV. Section V concludes with a discussion and an outlook concerning a potential implementation.

II. Context

For the extraction of detailed requirements for real-time hardware integration into the simulation of cyber-physical energy systems, adjunct applied use cases are subsequently listed. Furthermore, relevant actors (in the form of software or hardware/lab components) for those use cases are identified and their interaction is analyzed, respectively. This illustrates their mutual dependencies and helps with the identification of implicit systems.

A. Use cases for hardware interaction in CPES simulation

In the course of recently conducted research activities the following four use cases (UC), which are relevant to the intended development, have been identified. Within these research activities, an increased interest of industry partners for widely applicable co-simulation concepts for designing and analyzing cyber-physical energy system has been recognized, especially with a focus on the influence of communication systems and hardware interaction. Some of these research projects are summarized below.

`UC-A` – *Rapid control prototyping and evaluation for networked smart grid systems:* Active control of grid components and small-scale generators is one option to increase the hosting capacity for renewable energy resources in power distribution grids [15]. Control solutions are being developed in cooperative research projects, such as [16], but also in industry. Active intervention in the operation of the distribution grid can be realized with the help of tap changer transformers, adjustable (re)active power set-points of generators, or even topology changes of the distribution grid. The more distributed the sensors and actuators of such control systems are, the more challenging becomes testing during development and validation of such solutions. One approach is to incorporate highly specialized domain-specific simulation tools in a common co-simulation environment. A process for rapid control prototyping has been developed [17] that incorporates distribution grid and communication simulation as well as C-HIL evaluation for this purpose.

`UC-B` – Smart grid components communication and interaction tests: Interoperability of supply equipment for electric vehicles (charging stations) needs to be tested to assure that battery charging for a given electric vehicle is possible with any supply equipment (EVSE). The electric vehicle use case is challenging since it includes a number of different physical and communication interfaces that need to interact in a managed charging setting. A setup for testing supply equipment and its communication interface to the management system (using Open Charge Point Protocol — OCPP) has been proposed [18] in the course of the EU FP7 COTEVOS[1] project. The focus was on testing and certification of real world EVSE in the lab with a high number of simulated EVSE and under various EV charging scenarios.

`UC-C` – Smart grid power hardware device/stress tests: Before product release, smart grid devices such as controllers, protection systems, or inverters are tested for their robustness and behavior under various environmental conditions. These tests consider (high) power hardware-in-the-loop (P-HIL) evaluation under real world conditions (e.g., using a climate chamber, real-world voltage profiles, realistic fault situations). Future systems will require significantly more interaction with other devices (e.g., sensors or actuators elsewhere in the grid) as it is the case today. Therefore, close coupling of the power hardware in the lab with simulated physical systems is highly desirable for efficient testing.

`UC-D` – Evaluation of distributed multi-agent based control systems for ancillary service provision: Reliable ancillary service provision utilizing distributed energy resources (DER) can be realized with multi-agent systems (MAS), which take care of the optimization of the underlying hardware and configuration of the autonomous responses (typically from DER inverters). While the feasibility of the MAS implementation may be evaluated in purely software-based environments, the provision of power reserves itself is an autonomous reaction to grid parameters (e.g., droop controlled f/P-response) and thus needs be tested in a dynamic (preferably hardware) environment. However, stability of the integrated hardware/software system can only be guaranteed when providing an adequate environment [19].

All of the use cases listed above were developed within already completed or still ongoing research projects. They can be considered as specific instances of one generalized use case (`GUC`), as described below. This abstraction is used because (according to [20]) the more complex a system is, the more advantageous for its realization is the abstraction of use cases.

`GUC` – Real-time hardware integration into co-simulation of cyber-physical energy systems: This generalized use case covers `UC-A` to `UC-D`. Their common aspects are twofold. On the one hand, they all incorporate co-simulation concepts for (physical) energy systems and partially communication systems and controllers, which can be summarized as co-simulation of CPES. On the other hand, they all require the integration of real hardware components – either power or control hardware – into the simulation.

Due to its comprehensiveness and level of abstraction, all other use cases can be seen as specializations of `GUC`. Consequently, all the requirements derived from the individual use cases are also reflected by the generic use case `GUC`.

B. Relevant actors

Focusing on control of CPES and/or the evaluation of physical components, the following four actors can be identified. Each of these actors represents a set of software or hardware components that is relevant in CPES simulation.

- physical systems simulators
- communication simulators
- controls (algorithm prototypes/hardware)
- power hardware/real time digital simulators (RTS)

The physical systems simulators represent real world environments, e.g., power systems simulators such as PyPower, PowerFactory, NEPLAN, OpenDSS, or PSS SINCAL. They are primarily continuous systems simulators maintaining physical state variables like voltage, power, temperature, or energy. On the other hand, communication simulators are modeling the discrete behavior of (message/packet based) communication systems. Widely used communication simulators are the network simulator ns-2/ns-3, OMNeT++, and OPNET Modeler. Controllers for CPES can manifest as actors in two ways: either as prototypes, which might still be in the development phase, or as implemented controller hardware components, which are field-ready. The last actor of this list represents power hardware of any type and real time digital simulators (RTS), which operate at very dense time intervals and are able to physically emulate dynamic system behavior.

C. Different types of data to be transferred

Concerning data transfer among actors during co-simulation of CPES with real-world/lab hardware components, the following different types of data can be distinguished.

Simulation control signals are internally used for orchestrating co-simulation. They might start and stop simulation processes, communicate simulation scenarios, allow simulators to progress to a specific point in simulation time, and so on. Thus they can be seen as control flow signals.

Variables are (internal) states of a physics-based system models that influence other interacting physics-based systems

[1]EU FP7 COTEVOS project – http://cotevos.eu/

model simulated in other simulators and thus need to be communicated among those simulators.

Messages are explicitly issued by (the model of) an active element within a simulator representing sensors or actuators in the real world (e.g., voltage sensors, control elements). These messages are typically transferred via a specific protocol.

Which actors exchange what type of data is presented in the following section with the help of an actor interface diagram.

D. Actor interface diagram

Fig. 1 shows the actor interface diagram depicting the actors identified above for the generic use case (GUC) and their adjunct data types and paths.

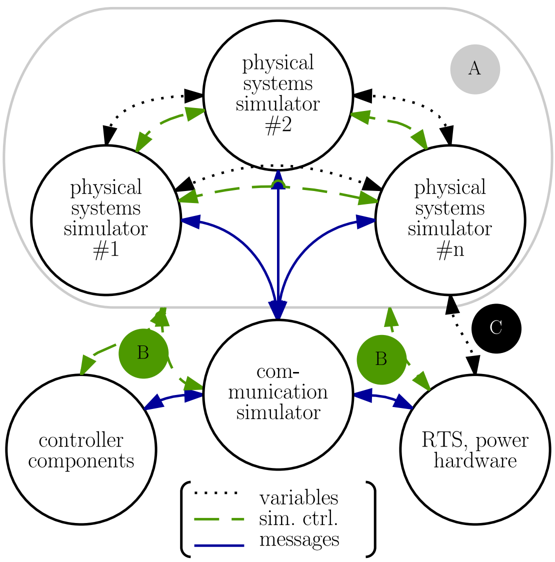

Fig. 1. Actor interaction diagram of physical simulators, communication simulator and hardware components; containing an implicit CPES system (A), optional simulation control signals (B), and optional physical variable exchange (C).

The topmost three actors represent physical systems simulators of which each one represents a specific physical domain. Each of these domains is represented by state variables, which dynamically change over time during a simulation. At least within the simulation of the CPES (A in Fig. 1) some of those variables – to be specific, those that influence the other physical system(s) – have to be exchanged (black dotted links). For the individual physical systems simulators to act jointly, some means of synchronization, common simulation time, and simulation control signals have to be exchanged. This coordination task can be managed by one of the physical systems simulators or a specific element (e.g., SimPy[2]), which is not explicitly included in the figure.

All actors within Fig. 1 are actively communicating via messages (e.g., meter readings from power grid simulations, set points from control systems to PV inverters, price values from a marked agent). These messages (blue solid links) are real world/field elements, in contrast to variables and simulation control signals. To model communication behavior of the real communication channel, communication simulators may be used.

The actors to the left and right of the communication simulator in Fig. 1 are lab components, which do not directly relate to the CPES software simulation. They are rather interacting components, which might be taken from the field. The actor on the lower left hand side represents controller components that typically only communicate via explicit messages to their counterparts in the field or the related simulated physical system. This communication typically takes place via (domain) specific communication protocols in a message-based discrete manner.

On the lower right hand side is the RTS/power hardware actor. This actor represents real time digital simulators (e.g., by OPAL-RT Technologies or RTDS Technologies) and power hardware components (e.g., on-load tap-changers, EV supply equipment). In contrast to the controller components, these components physically emulate continuous systems, which might also need to exchange (physical) variables from the lab to their representations in the CPES simulation (cf. C in Fig. 1). This physical state variable exchange between the lab and the CPES simulation is not mandatory but very likely, since the idea of hardware coupling lies in the close integration of those real world components into CPES simulation.

In case of hardware integration one more thing has to be considered, which is the integration of lab components into the simulation control flow. To fully integrate these real world components into a fully automated scenario-based simulation with a potentially altered time scale, also the lab related actors need to support simulation control flow signals (cf. B in Fig. 1). This again can be seen as optional but necessary for full simulation integration.

III. Description of the Generalized Use Case

This section discusses the generic use case (GUC) in an abstract, technology independent manner (schematically depicted in Fig. 2), which serves as the basis for the requirements extraction in the next section. The discussion is performed stepwise with increasing complexity, beginning with CPES simulation, which then is extended by communication systems simulation and finally enhanced by hardware coupling.

A. Simulation of cyber-physical energy systems

From a methodological point of view, the ability to properly simulate CPES is closely tied to the capability of handling hybrid models, i.e., models comprising both continuous time and event based properties. As mentioned above, in CPES this relates to the combination of continuous models of physical infrastructure (e.g., distribution grids, generation/storage units) with control decisions and other events related to energy systems operation (e.g., price signals, consumer behavior). From an engineering point of view, CPES are exceptional due to the fact that the modeling of use-cases typically requires a multi-domain approach, i.e., across the boundaries of traditional engineering domains (e.g., electrical grids, district heating, mobility). Since most energy-related simulation tools have a domain-specific focus, co-simulation concepts have become increasingly popular for designing and analyzing CPES.

[2]http://simpy.readthedocs.org/

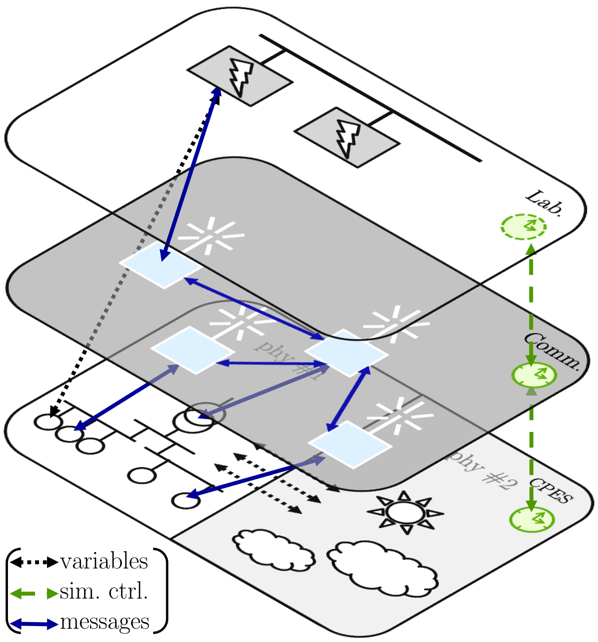

Fig. 2. Layer representation of the generic use case showing the CPES layer at the bottom, the communication simulation layer in the middle, and the lab hardware components on top; together with an example of potential interactions.

Consequently, a typical CPES in the context of the generalized use case `GUC` can be characterized as a hybrid multi-domain systems in the sense defined above (cp. with Fig. 1). This seemingly generic definition of a generalized CPES has indeed many implications for the practical implementation of simulation environments, for instance with regards to event detection/handling, determinism of execution or application programming interface definition. As outlined in Section I, several software-based approaches are currently available that aim at overcoming the associated technical challenges.

B. *Consideration of communication simulation*

Considering communication (and thus considering messages depicted as blue links in Fig. 2) is inevitable if one focuses on control system development, or deals with other actively operating or communicating elements (e.g., multi-agent systems). Messages are always issued by an active component or its model representation and can never directly arise from a physical simulation.

In case of ideal communication channels, messages might be directly forwarded from the originating actor to the target actor – comparable to the physical variables. However, to achieve realistic communication behavior, one or more communication simulators need to be incorporated. Including one such simulator means that all message exchange needs to be redirected over this actor (cp. with Fig. 1). As messages might origin from any actor – independent if it is hard- or software – they have to be collected, maintained, and forwarded by the communication simulator. This actor has to maintain one or more message queues (depending on the physical channel's nature) in order to guarantee determinism and reproducibility of the experiment. The message furthermore has to be considered by or synchronized with the CPES simulation.

Typically, communication simulators model temporal behavior (e.g., packet congestion, delay or transmission time) and thus need to be temporally coupled to all other actors with temporal behavior. As communication topology may vary from the physical system topology, it has to be separately considered within the communication simulation. Furthermore, a mapping between the actors – or even each of the communicating entities modeled by one actor – and the communication topology is necessary.

In the middle layer of Fig. 2 the communication simulation is considered. It represents the channels behavior and the communication system topology. Messages issued by models of actors in the CPES simulation are directly forwarded from the CPES layer to the communication simulation without any interference. Within the communication simulation it is treated according to the channel model (delayed, distributed to other nodes, altered, marked as lost/corrupt, etc.) and then returned to the destination actor models in the CPES layer or the real actors in the lab layer.

C. *Integration of lab infrastructure*

Simulation models of active CPES components can be fully considered within the CPES co-simulation consisting of the CPES layer and the communication simulation layer in Fig. 2. In contrast to that, the integration of real hardware components leads to challenges and necessary modifications and extensions of this system. Lab infrastructure has to be considered in both domains – considering their physical as well as their ICT interactions with the rest of the simulated system.

The physical integration has to be done by mapping of the hardware components' real physical state variables (e.g. terminal voltages, generated or consumed active and reactive power) to their representation within the CPES simulation. This has to be done for both input and output variables of the hardware component in an adequate way (i.e., conformity with sampling rate and accuracy requirements). In order to do so, controllable lab equipment (e.g., power amplifiers), generating the physical representations of the input variables, and measuring instruments (e.g., watt-meters), feeding back the response of the connected hardware's outputs into the CPES simulation, are necessary.

Integrating the communication-related part of the lab hardware also requires for a virtual representation of the lab hardware in the communication simulator to facilitate an interaction with other (simulated) entities. Consequently, a mapping of the hardware components' communication interface(s) to their virtual representations within the communication simulation is needed. This integrates the lab hardware into the communication topology.

IV. Requirements

Co-simulation of CPES and communication systems has already been performed in various ways [21]–[23] and is therefore assumed to be a solved problem (e.g., by the co-simulation framework mosaik [6]). This section focuses on the requirements that result from the integration of real power hardware and real time digital simulators (RTS) into such CPES co-simulation systems. Derived from the use case description in Section III – and especially Section III-C – the following requirements for a hardware coupling middleware can be identified.

A. Requirements for the CPES and communication systems co-simulation

To allow for hardware integration into the CPES simulation, the simulation framework – or to be more precise the CPES simulation assembled by the given co-simulation framework – has to fulfill the following requirements.

`R1` – *CPES simulation must allow for lab component integration concerning their physical state variables and messages:* The CPES simulation has to provide the means for lab component(s) to exchange and properly represent physical state variables and messages. This means that physical inputs and outputs of real hardware components have to be mapped to the respective component representations in the physical systems simulation. This also applies for the integration of messages from and to the lab hardware.

`R2` – *CPES simulation must allow for simulation control extensibility:* Events issued by real physical hardware, i.e., both updates of their physical variables and issued messages, have to be considered and included in the co-simulation framework's runtime execution. Furthermore, lab infrastructure must be included in simulation flow control (e.g., starting/stopping of the simulation, setting of initial parameters) and in scenario exchange. Thus, lab hardware infrastructure has to be handled by the simulation runtime control like another individual simulator with specific (timing) requirements.

`R3` – *CPES simulation must be real-time and wall-clock capable:* Hardware components are always bound to wall-clock time. Thus, CPES simulation must be capable of either operating in this mode or handling components that operate in this mode. Besides maintaining this pretended slowdown and linearization of simulation time, the CPES co-simulation framework must ensure to finish each simulation step at least within the simulated step duration, in order to avoid lagging behind the lab hardware components.

`R4` – *CPES simulation must provide variables in a timely and accurate way:* State variables of continuous dynamic simulations to other simulators can only happen at discrete points in time and with limited accuracy (quantization). The variable exchange has to be done frequent enough as to comply with the time scale of the lab hardware components' behavior.

`R5` – *CPES simulation must provide syntactical and semantic interface description:* For the interaction of the lab infrastructure an interface with a clear syntactic and semantic description has to be provided concerning all three types of data (cf. Section II-C) to be exchanged. This requires achieving syntactic and semantic interoperability by means of meta and reference models [24].

B. Requirements for the lab/hardware set-up

Hardware integration into CPES co-simulation includes a heterogeneous set of real world components from time continuous systems like power hardware and RTS to event based systems like controller units. For the following set of requirements the existence of a hardware abstraction layer is assumed, such as the software middleware proposed in [25] that maintains the integration of hardware components into an existing CPES co-simulation.

`R6` – *Consideration of heterogeneous hardware components:* A heterogeneous set of hardware components has to be considered for the integration into the CPES co-simulation. The increased complexity caused by this heterogeneous set of components, interfaces, and time bases has to be handled by means of abstraction and encapsulation. To allow for a reduction of complexity and to have a uniform interface between the lab set-up and the CPES simulation a hardware abstraction layer has to be created that encapsulates all the different hardware components.

`R7` – *Exchange of physical variables and messages:* Within the hardware abstraction layer, information about the contained hardware components and their attributes has to be maintained and provided consistent to a meta model defined by the co-simulation framework (cf. `R5`). Such attributes could be initial parameters that would have to be set prior to a simulation run or physical state variables which would have to be set and read during the simulation process. Furthermore, messages of the real hardware components need to be forwarded to the CPES co-simulation and vice versa (cf. `R1`). This exchange and the processing of these attributes has to be performed with sufficient performance by the hardware components to keep pace with the CPES simulation.

`R8` – *Conformity with simulation control signals:* The hardware components or their hardware abstraction layer, respectively, has to conform not only to the co-simulation framework's physical state variable and message exchange but also to its simulation control signals (as mentioned in `R2`). Furthermore, the special case that hardware components are operating in wall-clock time in comparison to the potentially different timing behavior needs to be considered (e.g., by dedicated synchronization measures).

C. Other requirements

`R9` – *Uncertainty quantification:* The unavoidable uncertainty that is caused by the coupling of both physical systems simulators and lab hardware components with their different time bases, models, and quantization effects has to be considered. Uncertainty estimation and the evaluation of the uncertainty impact to the whole setup are necessary to assess the overall system quality [26].

`R10` – *Reuse and extension of existing models and solutions:* For a successful implementation already existing solutions and established models should be used and extended by necessary features.

V. Discussion and Outlook

From recently conducted joint research project the need for real-time hardware integration into co-simulation concepts for CPES has emerged, especially from industry partners. In this paper the realization of this approach has been discussed and, as a first step, an established requirements engineering process based on a set of use cases derived from research projects has been applied. After abstraction of these use cases to a generalized use case the relevant actors have been identified, as well as their interaction, the type of data that has to be exchanged and the adjunct data paths. Furthermore, from a comprehensive description of the generalized use case, a set of basic requirements for the CPES co-simulation and the hardware components has been extracted.

In summary, it can be concluded that the framework used for the CPES and communication system co-simulation – which is a separate but solved task out of this work's scope – has to allow for integration of external components in the manner as for other distinct "simulators", with specific timing and sync behavior (due to its wall-clock operation and sampling of real physical signals). Exchange of both physical state variables and messages has to be provided by the CPES simulation framework (`R1`) and supported by the hardware (`R7`). The same (requirements) relation is true for the simulation control and synchronization. The hardware components have to act as another simulation client (`R8`), which needs to be supported by the framework (`R2`).

To act as a single simulation client, the heterogeneous set of hardware components needs to be unified and abstracted (`R6`). For this purpose, the introduction of a hardware abstraction layer for unifying all the different hardware interfaces has been suggested, allowing for seamless variable and message exchange with the CPES co-simulation framework and maintaining simulation control and synchronization of all real world components. Besides the unification of interfaces, such an abstraction layer creates modularity and thus allows convenient coupling and decoupling of the lab hardware components from the remainder of the CPES co-simulation. This would also allow for distributed operation of the simulation system beyond component, lab, organizational, and geographical boarders.

The introduction of a hardware abstraction layer raises the question of where to connect the communication simulator. On the one hand, it be considered as a component of the CPES simulation or as part of the CPES and communication co-simulation. On the other hand, as communication systems typically are not time continuous but rather discrete systems, they could also be seen closer to the lab components and thus connected to the proposed abstraction layer.

This question as well as the design and concrete implementation of such a hardware abstraction layer is subject of current work and will be presented in future publications.

ACKNOWLEDGEMENT

Parts of this work have been supported by the program *International promovieren in Deutschland – for all (IPID4all)* of the *German Academic Exchange Service (DAAD).*

REFERENCES

[1] T. Strasser *et al.*, "A Review of Architectures and Concepts for Intelligence in Future Electric Energy Systems," *IEEE Transactions on Industrial Electronics*, vol. PP, no. 99, pp. 1–15, 2014.

[2] M. Buescher *et al.*, "Integrated Smart Grid Simulations for Generic Automation Architectures with RT-LAB and mosaik," in 5^{th} *IEEE International Conference on Smart Grid Communications*, 2014.

[3] K. Mets *et al.*, "Combining Power and Communication Network Simulation for Cost-Effective Smart Grid Analysis," *IEEE Communications Surveys & Tutorials*, 2014.

[4] P. Palensky *et al.*, "Simulating Cyber-Physical Energy Systems: Challenges, Tools and Methods," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 3, pp. 318–326, Mar. 2014.

[5] J. Bastian *et al.*, "Master for Co-Simulation Using FMI," in 8^{th} *International Modelica Conference*, Dresden, 2011.

[6] S. Rohjans *et al.*, "mosaik – A modular Platform for the Evaluation of Agent–Based Smart Grid Control," in 4^{th} *IEEE PES International Conference and Exhibition on Innovative Smart Grid Technologies (ISGT Europe)*, 2013.

[7] H. Georg *et al.*, "Analyzing Cyber-Physical Energy Systems:The INSPIRE Cosimulation of Power and ICT Systems Using HLA," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2364–2373, Nov 2014.

[8] C. Molitor *et al.*, "MESCOS—A Multienergy System Cosimulator for City District Energy Systems," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2247–2256, Nov 2014.

[9] K. Anderson *et al.*, "GridSpice: A Distributed Simulation Platform for the Smart Grid," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2354–2363, Nov 2014.

[10] R. Isermann *et al.*, "Hardware-in-the-loop simualtion for the design and testing of engine-control systems," *Control Engineering Practice*, vol. 7, no. 5, pp. 643–653, 1999.

[11] D. Becker *et al.*, "An engineering environment for hardware/software co-simulation," in 29^{th} *ACM/IEEE Design Automation Conference*, 1992, pp. 129–134.

[12] N. S. Voros *et al.*, "Hardware/software co-design of complex embedded systems: an approach using efficient process models, multiple formalism specification and validation via co-simulation," *Design Automation for Embedded Systems*, vol. 8, no. 1, pp. 5–49, 2003.

[13] H. Morais *et al.*, "SOSPO-SP: Secure Operation of Sustainable Power Systems Simulation Platform for Real-Time System State Evaluation and Control," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2318–2329, Nov 2014.

[14] S. Rohjans *et al.*, "Requirements for Smart Grid Simulation Tools," in *IEEE* 23^{rd} *International Symposium on Industrial Electronics*, 2014.

[15] "Dena-Netzstudie II - Integration Erneuerbarer Energien in Die Deutsche Stromversorgung Im Zeitraum 2015–2020 Mit Ausblick 2025," Deutsche Energie-Agentur GmbH (dena), Berlin, Endbericht, Nov. 2010.

[16] H. Brunner, "DG DemoNet smart LV grid - increasing the DER hosting capacity of distribution networks - voltage control from simulation to field test," Vienna, Apr. 2012.

[17] M. Faschang, "Rapid Control Prototyping for Networked Smart Grid Systems Based on an Agile Development Process," Ph.D. dissertation, Vienna University of Technology, Vienna, 2015, (to be published).

[18] M. Nöhrer, *Flexible Co-Simulationsumgebung zum Testen von Ladeinfrastrukturen für Elektromobilität*. Vienna: Diploma Thesis, Vienna University of Technology, 2014.

[19] S. Lehnhoff *et al.*, "Distributed coalitions for reliable and stable provision of frequency response reserve," in *Intelligent Energy Systems (IWIES), 2013 IEEE International Workshop on*, Nov 2013, pp. 11–18.

[20] J. Kettenis, "Getting Started With Use Case Modeling: White Paper," Oracle, Tech. Rep., May 2007.

[21] H. Lin *et al.*, "Power system and communication network co-simulation for smart grid applications," in *2011 IEEE PES Innovative Smart Grid Technologies (ISGT)*, 2011, pp. 1–6.

[22] A. T. Al-Hammouri *et al.*, "Co-simulation tools for networked control systems," in *Hybrid Systems: Computation and Control*. Springer, 2008, pp. 16–29.

[23] T. Godfrey *et al.*, "Modeling smart grid applications with co-simualtion," in *2010 First IEEE International Conference on Smart Grid Communications (SmartGridComm)*. IEEE, 2010, pp. 291–296.

[24] M. Uslar *et al.*, *Standardization in Smart Grids: Introduction to IT-Related Methodologies, Architectures and Standards*. Springer, 2013.

[25] R. Mosshammer *et al.*, "Loose coupling architecture for co-simulation of heterogeneous components," in *Industrial Electronics Society, IECON 2013 - 39th Annual Conference of the IEEE*, Nov 2013, pp. 7570–7575.

[26] S. F. Wojtkiewicz *et al.*, "Uncertainty Quantification In Large Computational Engineering Models," in *AIAA Applied Aerodynamics Conference*, 2001.