

Research Article

A Cosimulation Architecture for Power System, Communication, and Market in the Smart Grid

Markus Mirz ¹, **Lukas Razik** ¹, **Jan Dinkelbach** ¹, **Halil Alper Tokel** ²,
Gholamreza Alirezaei ², **Rudolf Mathar**², and **Antonello Monti** ¹

¹*Institute for Automation of Complex Power Systems, RWTH Aachen University, Aachen, Germany*

²*Institute for Theoretical Information Technology, RWTH Aachen University, Aachen, Germany*

Correspondence should be addressed to Markus Mirz; mmirz@eonerc.rwth-aachen.de

Received 24 November 2017; Accepted 30 January 2018; Published 28 February 2018

Academic Editor: João Soares

Copyright © 2018 Markus Mirz et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Smart grids evolve rapidly towards a system that includes components from different domains, which makes interdisciplinary modelling and analysis indispensable. In this paper, we present a cosimulation architecture for smart grids together with a comprehensive data model for the holistic representation of the power system, the communication network, and the energy market. Cosimulation is preferred over a monolithic approach since it allows leveraging the capabilities of existing, well-established domain-specific software. The challenges that arise in a multidomain smart grid cosimulation are identified for typical use cases through a discussion of the recent literature. Based on the identified requirements and use cases, a joint representation of the smart grid ecosystem is facilitated by a comprehensive data model. The proposed data model is then integrated in a software architecture, where the domain-specific simulators for the power grid, the communication network, and the market mechanisms are combined in a cosimulation framework. The details of the software architecture and its implementation are presented. Finally, the implemented framework is used for the cosimulation of a virtual power plant, where battery storages are controlled by a novel peak-shaving algorithm, and the battery storages and the market entity are interfaced through a communication network.

1. Introduction

The increase of distributed energy resources (DERs) in power systems and the resulting bidirectional power flows are driving changes in the associated communication infrastructure and market mechanisms. For instance, the ongoing extension of measuring infrastructure in lower voltage layers requires a concurrent deployment of a capable communication network in order to provide for a reliable communication among control centers, substations, and measurement devices. Hence, the planning of electrical grid operation and the underlying communication network should be considered simultaneously, in order to enable the analysis of the impact of interactions between the two domains as it is already shown in [1]. Meanwhile, new market models are developed to support customers taking a more active role in their exchange of power with the grid [2] in such a way that their behavior will also be taken into account in the grid operation [3]. For example, individual users can contribute to a more efficient

operation of the grid by putting their battery storage at the disposal of the grid operators, which would require a communication network for the data exchange. Bearing this development in mind, it is interesting to include a market simulation in the analysis as well.

This integration of market mechanisms, the communication network, and the power system complicates studies on future power systems' behavior since a common modelling approach that encompasses the three domains has not been established yet and there are few tools that enable a joint simulation. The comprehensive data model and the cosimulation architecture presented in this work tackle these challenges, enabling the investigation of dynamic interactions between the electrical grid, communication network, and market. These interactions could be technical constraints to the grid that require actions on the market side, communication failures that affect the control loop between the grid and the market, and market decisions that change the behavior of a generation unit or energy consumers connected to the grid.

Therefore, we propose a data model based on the IEC Common Information Model (CIM) [4] that is able to describe an entire smart grid topology including communication and market entities. Besides, the proposed data model allows users to store the whole network topology with components from the three domains in a single well-defined data model, therefore hiding the complexity of the cosimulation from the users. Topology descriptions compliant with this data model can be processed by the presented cosimulation architecture. This architecture and implementation example combines dedicated simulators for the power system, communication network, and energy market whereas previous approaches known to the authors only considered a subset of these three domains or monolithic concepts.

The main contributions of this paper can be summarized as follows:

- (i) Analysis of the requirements of a cosimulation combining the three domains and definition of the cosimulation specifications
- (ii) Identification of technical challenges of interconnecting the simulators
- (iii) Development and the implementation of the cosimulation architecture and the interfaces that implement the specifications
- (iv) Validation of the proposed cosimulation environment by simulation results.

The paper is structured as follows: Section 2 gives an overview of the related work on integrated modelling and simulation of smart grids. In Section 3, we identify and describe the use cases for which the proposed cosimulation environment can be used. The challenges for the realization of the cosimulation environment are discussed in Section 4, whereas we present our solutions regarding the designed data model and the software architecture but also its limitations in Section 5. The integrity of the cosimulation environment is validated by the results of a cosimulation in Section 6, where an optimal management of distributed battery storage systems is simulated to improve the voltage stability of a distribution network. We conclude the paper with final remarks in Section 7.

2. Related Work

2.1. Architecture and Data Model. The evolution of traditional power systems towards intelligent power grids has recently triggered efforts for comprehensive modelling and standardization, which aim to include various aspects of a smart grid ecosystem. One major contribution in this context is the *Smart Grid Architecture Model (SGAM)*, which provides a basis for the representation of relationships between entities, functionalities, and subsystems in smart grids [5]. SGAM framework models a smart power grid in five interoperability layers which cover physical components in the network (*component layer*), protocols for exchange of information between services or systems (*communication layer*), data models which define the rules for data structures

(*information layer*), functions and services (*function layer*), and business and market models (*business layer*). The model further divides the component layer to hierarchical electrical energy conversion and transmission *domains* from *generation* to *customer premises* and to component *zones* such as *field* and *station*. This architectural view aims to accelerate and standardize the development of unified data models, services, and applications in industry and research. In this context, our data model and cosimulation framework in this work build upon the fundamental concept of SGAM, where our focus lies in the following aspects of SGAM:

- (i) The unified data model which is presented in Section 5.1 formally defines the structure for data exchange in alignment with the concept of the *information layer* of SGAM.
- (ii) The domain-specific simulators of our cosimulation environment include models of power system and communication network components as well as market actors in the *component layer* of SGAM within the *distribution*, *DER*, and *customer premise* domains.
- (iii) The communication layer is abstracted by a cosimulation interface and the extensions in the domain-specific simulators in order to enable the data exchange between the components.
- (iv) The example use case presented in Section 6, regarding the optimal management of distributed battery storage systems, is an example of a system function which would fall on the *function layer* in the SGAM framework. Besides, the business model that motivates the provision of such a system function, for example, an incentive by the system operator, is defined within the *business layer*.

For the realization of the unified data model in alignment with the SGAM concept, we identify the IEC Common Information Model (CIM) [4] as a well-established basis for extensions in Section 5.1 as it is a widely accepted format to exchange grid data in power systems. Although it includes an extensive list of electrical grid components as well as market-related objects, the communication infrastructure of a power system is hardly included except for a few classes regarding supervisory control and data acquisition (SCADA) links. However, CIM can easily be extended as demonstrated for the market in [6, 7].

2.2. Simulation of Smart Grids. There have been several attempts to build a cosimulation environment with the focus on power grids and communication networks, for example, [1, 8–10]. However, the proposed approaches so far do not account for the market because they focus more on short-term effects caused by limitations of the communication network.

Although MOCES [3] attempts to take a holistic approach to model distributed energy systems, this results in the implementation of a monolithic simulation instead of a cosimulation, with a hybrid simulation for the physical part and an agent-based simulation for the behavioral part which could

represent the market. Therefore, this approach hinders the use of existing domain-specific tools and requires substantial development effort. Instead, in this work, we aim to take advantage of the capabilities that existing tools offer which (i) enhances the credibility of the results and (ii) decreases the effort that is needed to replicate their functionalities.

The proposed simulation framework consists of several simulators, each responsible for a specific domain. The advantage is the possibility of using the best tool for each domain. Besides, the vision is to be able to replace simulators with reasonable effort if the simulation requirements change. For example in Section 6, it is mentioned how [11] could be used to implement a more comprehensive market simulation.

2.3. Classification of Simulations. Schloegl et al. [12] provides a clear classification scheme for energy-related cosimulations. Because our cosimulation environment shall provide holistic simulations, all four categories of simulation elements or models (i.e., continuous processes, discrete processes and events, roles, and statistical elements) defined by Schloegl et al. have to be considered.

In our implementation, the power system is modelled in Modelica as it can be used for physical systems in general [3] and has already proven its capabilities for thermal systems [13] and power systems in particular [14]. Modelica models consist of continuous processes, discrete processes, and events, which makes the power system simulation a hybrid simulation. The communication network is simulated with the available discrete event simulation (DES) tools, such as ns-3. In a DES, the simulation time proceeds with the execution of single events, such as packet arrival and time expiry [15]. The energy market simulation is implemented in Python as DES. Python was chosen as programming language because of its suitability to implement and test different optimization methods [16]. Each market participant aims at optimizing the schedule for its assets, for example, minimizing energy costs and maximizing its profit. Depending on the simulation scenario, the behavior or role of the market participants changes. Examples for statistical elements are wind farm models of the energy grid simulator and communication link models which consider packet losses and the reliability metrics of the communication network simulator.

Furthermore, the proposed cosimulation environment can be formalized as a coupled Discrete Event System Specification (DEVS) as defined in [17] which is why a classification of our architecture in terms of DEVS is given in Section 5.5.

3. Use Cases of the Cosimulation Environment

Our cosimulation solution including the three domains can be used to evaluate different scenarios. To address the physically relevant dependencies, we divide the dynamic interactions between the simulators into two groups:

- (i) Fast phenomena in the range of microseconds to seconds, between highly dynamic power system components, for example, power electronics and communication network

- (ii) Slow phenomena in the range of minutes to hours which include market entities, power system, and communication network.

In recent literature, the most prominent fast dynamic interactions occur in wide area measurement and control applications as well as remote power electronics devices [10]. Compared to slow phenomena, the simulation of fast phenomena requires relatively smaller simulation time steps due to switching events of electronic components and the fact that the market interactions are not considered. Therefore, in a cosimulation of the fast phenomena, the focus should be more on the efficiency of the cosimulation interface rather than a simplification of the coupling with multiple simulators. The proposed cosimulation environment is intended to support the investigation of fast dynamics but should not be limited to them. This has an influence on our design of the communication interface between the simulators as can be seen in Section 5.4.

On the other hand, slow phenomena are created by closing the loop between the power system and market by means of the communication network. The behavior of market entities may cause changes in the usage of power system components which has an impact on the grid and vice versa and this loop might be impaired by the communication network. The cosimulation environment might be used either to verify the functioning of the control loop given a specific communication network or to plan a communication network that fulfills the requirements of the control mechanisms in power system and market.

Based on this classification of use cases, it can be concluded that all three simulators do not necessarily have to be active at the same time for all scenarios. In this paper, we will focus on the requirements and the implementation for the cosimulation of slow phenomena and confine our discussion on fast phenomena to a description of the adaptations needed for fast phenomena investigations since there already exist several cosimulation environments proposed for these studies [1].

An example use case for slow phenomena is the optimal management of distributed storage systems for peak-shaving to support the grid operation. The proposed cosimulation environment including the communication network allows testing the effects of communication failures on the operation strategy and eventually on the electrical grid, which can provide valuable insights for decision-making. Simulation results for this example are provided in Section 6.

Before the cosimulation is initiated, it is necessary to define and store the topology under investigation along with the scenario-specific parameters. For example, the scenarios can be investigated in which the failures in the communication network are stochastically or deterministically set by the user in the data model. From a user perspective it would be advantageous if all components, their links, and parameters could be defined in one environment rather than splitting this information between different software solutions and formats. Then, the data model for the topology needs to include components that couple different domains.

From these use cases, the following challenges can be identified:

- (i) Common data model that includes components of all domains and their interconnections
- (ii) Interaction of simulators with different simulation types, for example, event-driven for the communication network and continuous processes for the power system
- (iii) Choice of the cosimulation time step which is limited by the synchronization method connecting the simulators.

4. Challenges of the Cosimulation

4.1. Common Data Model. A common data model that covers the electrical market, communication infrastructure, and grid does not exist to the best of our knowledge even though these components are integral part of smart grids. Not only would the user of a simulation software benefit from a common data model during the specification of the simulation scenario but the data exchange would also be simplified. A system description that encompasses all components of smart grids as shown in Figure 1(a) could be either used directly by holistic smart grid simulators or divided into subsystems for a cosimulation as in Figure 1(b).

For many components, this division is straightforward since their parameters are only needed by one domain-specific simulator. For example, electrical lines exist only in the power system domain and have connections only to other power system components. Some components, on the other hand, constitute natural coupling points between the power system, the market, and the communication network. These components are called *interdomain components* in the following. For instance, a battery storage device connected to the grid can act as a market participant that offers its capability to charge or discharge. In order to enable its participation in the energy market, the battery storage needs an interface which is a communication modem in this case. The modem can be seen as a part of the battery storage. Therefore, the data model class associated with the battery storage device has to be able to hold or reference to data on the properties of the battery storage in electrical, market, and communication domains. For a cosimulation, the information on interdomain components have to be split into several parts since their parameters are needed by the simulators and each simulator has to simulate a dedicated part of these components.

4.2. Cosimulation Time and Synchronization. A remaining and major issue in such a cosimulation, in which simulators of different categories are combined, is the selection and implementation of a proper synchronization mechanism which must ensure the proper progress of the simulation time and a timely data exchange between the domain-specific simulators. This selection is of crucial significance in order to minimize the error propagation in the cosimulation and the synchronization overhead in terms of simulation time. In [1], three main synchronization methods for cosimulation

have been specified: time-stepped, global event-driven, and master-slave. There are two different considerations related to the simulation time [12]:

- (i) Time resolution: a challenge of the cosimulation is the highly diverse time resolutions of the three simulators. The time steps of power grid simulator reach from milliseconds (electromagnetic processes) to subseconds and above (steady-state and electromechanic processes). The time steps of communication network simulations can even vary from tens of microseconds (e.g., latencies in LANs) to seconds (e.g., latencies in WANs). On the other hand, in energy market simulations, a time step of several minutes can be sufficient as it is the case for Germany's control power market with price calculations on 15-minute basis.
- (ii) Time ratio: time ratio describes the relation between simulation time and wall clock time [12]. With appropriate use cases, we want to show how holistic cosimulations with our approach can be used for planning and decision-making. Therefore, it is important to run the scenarios much faster than wall clock time and for time intervals of days as well as of weeks, so that many models with different configurations can be simulated.

The problem is that a very small time step or a high resolution in time impedes short simulation times. Therefore, it is necessary to adjust the time step according to the phenomena that are under investigation. At the same time, it may be feasible to aim at a higher integration of the simulators and sacrifice flexibility and increase the efficiency if both time resolution and time ratio need to be optimized. In Section 5.3, we discuss our design choices regarding the synchronization of the simulators in detail. The approach for an optimization of time resolution and time ratio is further discussed in the Section 5.4.

5. Concept of the Cosimulation Environment

5.1. Data Model for Power System, Communication Network, and Market. As explained in Section 2, one possibility is to extend the cosimulation topology format on CIM, which leads to a superset of CIM. New classes, which are introduced for completing CIM in its representation of smart grids, are linked to the CIM classes using the terminology of the Unified Modeling Language (UML). The proposed format can be structured in four packages:

- (i) Original CIM (IEC61970, IEC61968, IEC62325)
- (ii) Communication
- (iii) Market
- (iv) EnergyGrid.

Whenever possible, the CIM classes are used. However, some components might not have an associated class in the standard yet. Then, these components are represented as classes in one of the other three packages. The reason for

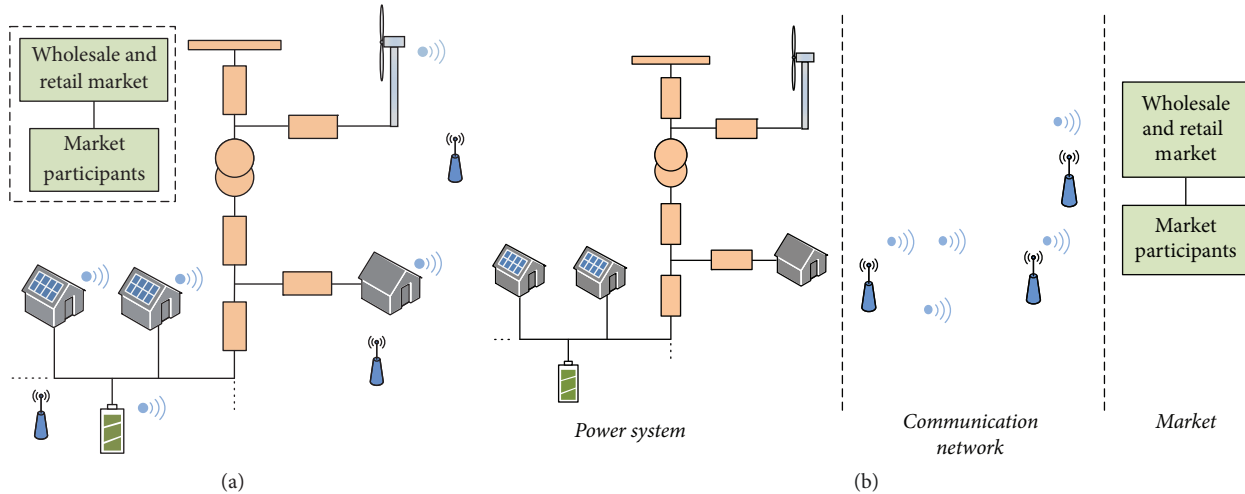


FIGURE 1: Exemplary topology including components of all domains (a) and domain-specific topologies (b).

this approach is that this way it is easier to update to a new version of CIM without losing the newly added classes and their interconnections.

The most important feature of our model format is the interconnection of domains. In order to accomplish this, we have identified possible interdomain components, namely, `BatteryStorage`, `SolarGeneratingUnit`, and `MarketCogeneration`, are shown in Figure 2, which is an excerpt from our data model. According to the UML diagram, the energy market components are associated with the power system components, whereas power system components have an aggregation relationship to communication devices. This means that parameters specific to the market, communication network, and power system which relate to the same device are linked with each other. Therefore, all information on one device is easily accessible but at the same time there is a separation according to the domains. The connections between classes of different domains are defined in a logical and not a topological way. Instead, topological connections exist to connect power system components, for instance.

Coming back to the battery storage device example, the data model is as follows: the device is a part of the grid and has electrical parameters. Furthermore, the battery storage might participate in the market, for example, as part of a virtual power plant (VPP). Market-specific information can be stored in the `MarketBatteryStorage` class which is associated with the electrical representation `BatteryStorage`. The data on the class for a communication modem `ComMod` which could be used to communicate with the VPP is aggregated to the `BatteryStorage` class.

In the following, we briefly mention the domain-specific considerations for the three domains.

(1) *Power System Package.* The purpose of the `EnergyGrid` package is to group models for power system components that are not already part of the CIM standard. For the simulation scenario that is presented later, it was necessary to create

a new model for electrical energy storages like stationary batteries. A battery storage is a conducting equipment that is able to regulate its energy throughput in both directions. Therefore, the class `BatteryStorage` is a specialization of a CIM `RegulatingConductingEquipment` since it can influence the flow of power at a specific point in the network.

(2) *Market Package.* The key component of the market package for the scenarios that we would like to investigate is a VPP since the aggregation of small DER units enables their participation at electricity markets. In case the DERs are not owned by the operator of the VPP, they can be seen as customers that offer their energy in exchange for share of the VPP operators profits. Besides, a VPP might support the Distribution System Operator (DSO) in ensuring a safe grid operation which results in an association between the DSO and the VPP.

(3) *Communication Package.* This package includes all additionally defined classes that are related to the communication network model, such as classes for communication links and technologies, modems, network nodes along with their parameters and their relations with the classes in CIM, power system package, and market package. In this way, the user can model the communication network layer and its specifications in the simulation tool. The communication network topology and its parameters are then used by the communication network simulator for the cosimulation.

Figure 3 shows an excerpt from the communication data model with an aggregation to a `WindGeneratingUnit`. By means of the associated classes for modems, communication requirements, and channels, the model enables a description of network parameters and topology. Furthermore, the communication network data model integrates the flexibility to enable the planning of the communication network under communication and power system requirements even before the beginning of the cosimulation. The topological parameters will be used for an optimal design of the communication network with the desired objective, such as minimum

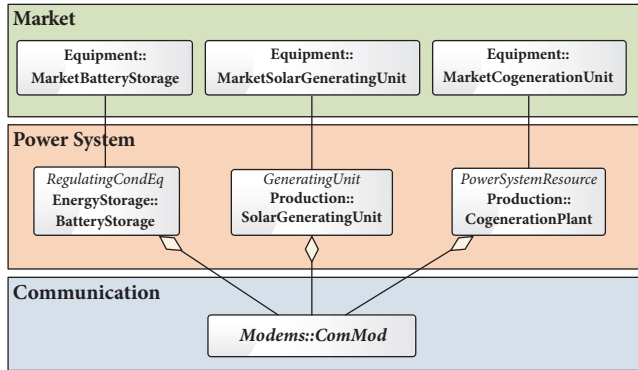


FIGURE 2: Interdomain connections between classes of power system, communication network, and market.

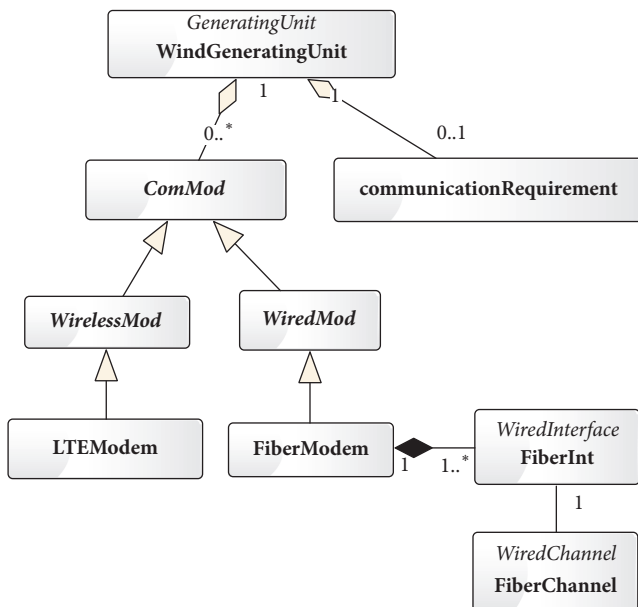


FIGURE 3: Communication network class association example.

cost, while required conditions are satisfied, such as given reliability metric. An example of this approach is presented in [18] for an integrated design of a wide area measurement system. Eventually, the optimized communication network solution can be evaluated against different scenarios using the cosimulation environment.

5.2. Model Data Processing and Simulation Setup. The overall information flow for the simulation setup is depicted in Figure 4. After the topology is created or modified in a graphical Topology Builder and includes the three domains, the input file, which complies with the common data model of Section 5.1, is sent to the cosimulation interface. The cosimulation interface incorporates a component, based on CIM++ [19], which parses the CIM XML-RDF file and generates a container of C++ objects that contain the topological data. In order to execute a simulation, the Modelica solver requires a Modelica model, whereas

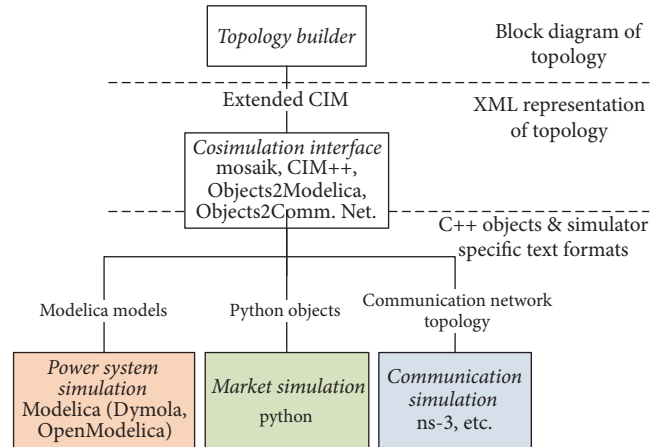


FIGURE 4: Overall architecture for simulation setup.

the communication network topology can be given to the communication network simulator in JSON or XML format, which includes the components in the network, their connections, and parameters. Since the topology will be available as an XML-RDF file and a container of C++ objects, the relevant information for the power system and communication network is extracted during a deserialization step. In the subsequent transformation step, a component, which we call Objects2Modelica/CommunicationNetwork, generates Modelica and communication network files with the topology and parameters. In contrast, the Python based market simulation relies on a C++/Python interface, which could be realized using one of the common libraries for Python to wrap C++ data types and functions, to retrieve the market relevant information from the C++ objects and store them in Python objects.

The following paragraph explains the translation on the basis of a CIM to Modelica example. The loads are defined in the extended CIM data model described in Section 5.1 as PQ-loads with a characteristic power demand as follows:

```
<cim:SvPowerFlow rdf:ID="PQ1-sv">
<cim:SvPowerFlow.p>1000</cim:
SvPowerFlow.p>
<cim:SvPowerFlow.q>329</cim:
SvPowerFlow.q>
<cim:SvPowerFlow.Terminal
rdf:resource="#E-1229789360"/>
</cim:SvPowerFlow>
```

The cosimulation interface extracts the corresponding component parameters from the CIM XML-RDF file and introduces them in the Modelica model of the grid. The latter is done by the Objects2Modelica component explained in Section 5.2 which iterates through the list of C++ objects provided by CIM++. Thus, the specification of the parameters P and Q in the CIM model generates two attribute modification equations in the declaration of the PQ-load in Modelica:

```
ModPowerSystems.PhasorSinglePhase.Loads.
PQLoad CIM_PQ1 (Pnom = 1000, Qnom = 329)
```

These values are applied during the quasistationary simulation of the single-phase representation of the grid.

5.3. Synchronization. The synchronization of all three simulators will be performed in fixed time steps. Fixed synchronization time steps have been chosen because of the resulting flexibility to integrate more simulators easily and its comparatively high speed in terms of time steps per simulation time [1]. An event-driven approach as it is implemented in [10] for two simulators requires a deeper integration of the cosimulation framework and the simulators. Apart from that, it was shown in [10] that the cosimulation error can be reduced for fixed time steps by reducing the global time step size.

The synchronization between all simulators for slow phenomena scenarios is performed with mosaik, a well-established cosimulation framework [20] which was developed for stationary simulations with time steps of one second or greater [21]. It allows combining the three simulators in a simple manner as explained in Section 5.4. VILLASnode, a software project for coupling real-time simulations in LANs [22, 23], is a suitable alternative for mosaik in the case of synchronizations with very short intervals.

In Modelica, the synchronization data exchange is achieved by integrating Modelica blocks of the Modelica_DeviceDrivers library, which was originally developed for interfacing device drivers to Modelica models. This conveniently allows the definition of a fixed interval for data exchange. Modelica_DeviceDrivers were chosen instead of the FMI approach shown in [21] because it allows more flexibility in picking the desired simulation variables and choosing the required cosimulation step size independently from the Modelica simulator time step. At synchronization steps between all simulators, the step function of the energy market simulator is called synchronously from the mosaik Python API. After the market simulator has finished the simulation step, the results are retrieved by mosaik.

The simulation time in the DES of the communication network advances with the execution of the generated events, which are stored in an event-list. The execution of events is controlled by a scheduler, which determines the next event and its execution time. Whereas the default scheduler executes the events sequentially without any interruptions, the available simulation environments offer the flexibility to integrate external control inputs to receive external control messages, which can be used to manipulate the simulation flow by changing the module parameters during the simulation. In order to realize this, the event-driven nature of DES tools can be used to generate new events, called flow control events, which stop the communication simulation, exchange data, and use the input data to manipulate the following simulation steps. Furthermore, the execution of events can be controlled and the simulation can be stopped after the execution of the last event before a synchronization point, so that the simulation runs in fixed steps and a data exchange is possible at the end of each step.

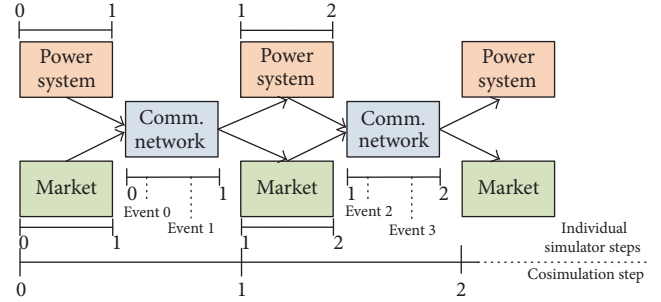


FIGURE 5: Synchronization scheme of simulators at cosimulation time steps.

Figure 5 depicts the flow of time for the cosimulation and each simulator. It can be seen that the power system and market simulators compute in parallel, whereas the communication network is waiting for their inputs. In a mathematical notation, the interactions between the simulators in each cosimulation step can be defined by

$$\begin{aligned} u_p(n+1) &= F_c(F_m(u_m(n))), \\ u_m(n+1) &= F_c(F_p(u_p(n))), \end{aligned} \quad (1)$$

where u_m and u_p are the corresponding input values of the simulators for the power system, energy market, and communication network for each time step. Therefore, it is required to set initial values, $u_p(0)$, $u_m(0)$, at the beginning of the cosimulation. n denominates the current cosimulation time step. F_c (communication), F_m (market), and F_p (power system) are the functions describing the calculation of the next time step.

5.4. Cosimulation Runtime Interaction. In Figure 6, the coupling of the power system, communication, and market simulator for their cosimulation runtime interaction is shown. In the following, we briefly introduce the individual parts of the cosimulation environment:

- (i) Mosaik: as already mentioned, mosaik is used for the coordination during the synchronization steps of several minutes (in simulation time) regarding all simulators [24]. mosaik has two different APIs for simulator coupling. It provides handlers for different simulator types, allows modelling of different simulation scenarios, and schedules the step-wise execution of the connected simulators with the aid of SimPy [25]. The two mosaik APIs are
 - (a) the low-level API which uses common TCP/IP network sockets for exchanging messages encapsulated in JSON, an open-source and human-readable data format;
 - (b) the high-level API which can be directly used by a simulator and communicate with mosaik through sockets but also handle their creation, events, and message (de)serialization.

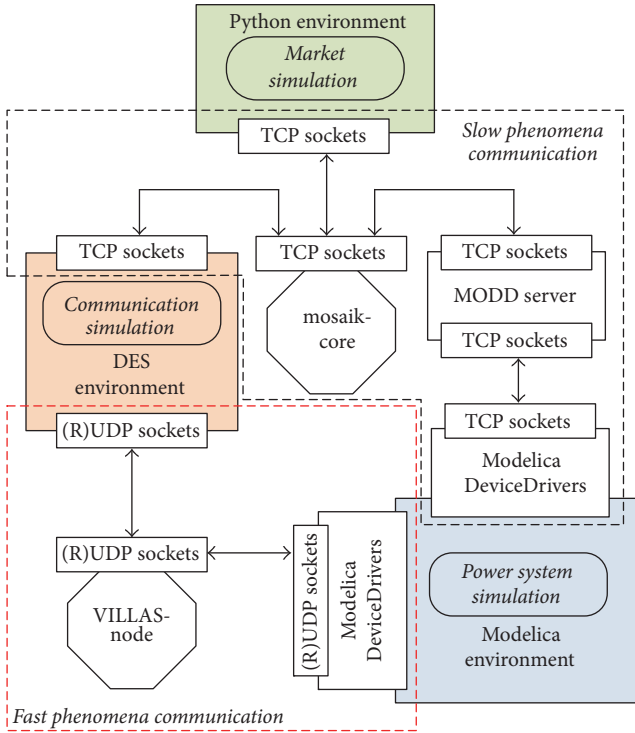


FIGURE 6: Scheme of runtime interaction between cosimulation components.

- (ii) Market simulator: implemented in Python, it can make use of the high-level API as illustrated in Figure 6. The use of the sockets allows the desired flexibility of running all simulators on different computer systems and environments.
- (iii) Communication network simulator: based on available DES tools, their network simulation modules are extended with interprocess communication functionalities for JSON message exchange with mosaik.
- (iv) Power system simulator: the integration of so-called *TCPIP_Send/Recv_IO blocks* from *Modelica_DeviceDrivers* into the Modelica models, allows the exchange of simulation data via sockets but in the form of Modelica variables as bitvectors instead of JSON messages. Therefore, the MODD Server is implemented.
- (v) MODD server: it receives commands from the socket connected with mosaik. Based on these commands it starts, for example, the power system simulator or receives the bitstream from *Modelica_DeviceDrivers* and encapsulates it into JSON messages before transferring them to mosaik. Besides the synchronization steps controlled by mosaik, there will be also more fine-grained synchronization steps of fractions of seconds between the power system and communication network simulator. That is why a VILLASnode server is included.

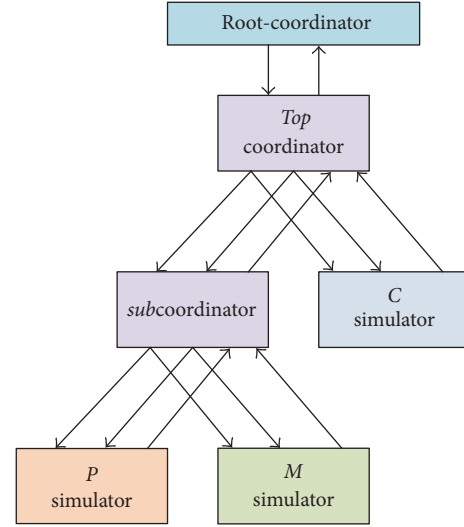


FIGURE 7: Cosimulation environment as a hierarchical simulator.

- (vi) VILLASnode: instead of TCP (Transmission Control Protocol), it makes use of UDP (User Datagram Protocol) sockets for data exchange between real-time simulators, for which it was designed. The use of UDP leads to less overhead and consequently to lower synchronization time steps but with the disadvantage of an unreliable connection. Our solution for avoiding any datagram losses is the usage of Reliable UDP (RUDP) to keep a low overhead in comparison with TCP with the benefit of a reliable connection.

5.5. *DEVS Formalization.* As explained in Section 5.3, a discrete time base is chosen for the synchronization between the domain-specific simulators. Therefore, the simulators can be formalized by Discrete Time System Specifications (DTSS). Since the time steps are fixed, the DTSS can be simulated by DEVS [17]. The abstraction level of the DEVS has not been considered for the mosaik framework for different reasons [26] and is not needed for the definition of cosimulation scenarios. Therefore, a formal definition of the whole simulation as coupled DEVS is beyond the scope of this paper. However, a so-called *hierarchical* simulator for the three *atomic* simulators *P*, *M*, and *C* (for energy, market, and communication) is shown in Figure 7 as leaves of the hierarchical simulator (i.e., tree) [17] and described in the following. The *root-coordinator* sends an *initialization message* (i, t) , which is propagated to all atomic simulators in the tree at simulation start and initiates the simulation steps with *state transition messages* $(*, t)$. The *coordinators* handle the levels of coupled simulators up to the root of the tree. The scheduling of an event is performed by a $(*, t)$ message forwarded by each receiver coordinator to its *imminent child*. The imminent child is the simulator which shall perform its next *internal transition* or the coordinator being the root of the subtree containing the simulator which performs the next internal transition.

In case of our cosimulation architecture, the first $(*, t)$ message is forwarded by the *top* and the *subcoordinator* to the first component in the *event-list* (here. P simulator) computing the new output $y_P = \lambda(s)$ and sending it as *output message* (y_P, t) to the subcoordinator. The subcoordinator then recognizes an *internal coupling* and therefore sends an *x-message* (x_M, t) , with $x_M = Z_{P,M}(y_P)$, to the M simulator, whereby $Z_{P,M} : Y_P \rightarrow X_M$ is the *translation function* from the output events of the P simulator to the input events of the M simulator. Although the computations of the M simulator, within the same transition, do not depend on the output of the P simulator, its output message (y_M, t) includes the output of both simulators. This output is translated and forwarded by the subcoordinator to its parent (top coordinator), because of the *external coupling*, as output message (y_N, t) of the belonging Discrete Event Specified Network (DEVN). Afterwards, the top coordinator translates the message to an input message (x_C, t) of the C simulator which computes the output of the whole cosimulation step. This output is transformed by the top coordinator, because of its internal coupling, to an input message for the subcoordinator which translates and sends the message down to the P simulator, because of the *external input coupling*, and the next cosimulation step begins.

An improvement of this formalization based on the hierarchical simulator could be accomplished by a formalization of the P and M simulator as a *conservative parallel discrete event simulation* [17]. Nevertheless, the described communication pattern between the components of the hierarchical simulator shows that the chosen cosimulation synchronization scheme, depicted in Figure 5, is valid, when causality violations between the energy and market simulation within one time step of the cosimulation are avoided which is guaranteed for the considered simulation scenarios.

5.6. Limitations. Due to the communication overhead caused by the cosimulation environment, the simulation time might increase significantly for large network sizes. Currently, real-time simulations are not possible, but the available framework can be extended for real-time and hardware-in-the-loop simulations, for example, by using VILLASnode instead of mosaik.

Furthermore, the synchronization step size cannot be changed during simulation runtime, whereas the domain-specific simulators support variable simulation time steps. To enable a variable step size for an optimized cosimulation through more sophisticated algorithms, modifications of the mosaik framework would be necessary as it allows fixed time steps only. The cosimulation flow managed by mosaik, which allows parallel and sequential progress of simulators, might introduce inaccuracies in the simulation results with respect to the synchronization step size.

Moreover, the simulation of heterogeneous communication networks and standards is possible. However, the abstraction level of the communication protocols influences the simulation time of the communication network simulator and thus the simulation time of the cosimulation.

6. Verification of the Cosimulation Interfaces

In this section, we aim to show that the implemented cosimulation infrastructure does not impair the accuracy of the simulation results. As an exemplary application, we consider a scenario where a VPP operator aims at gaining profits by reducing the VPP's peak power. Thus, a peak-shaving algorithm is employed for an optimal management of distributed battery storage systems. Financial incentives for the peak-shaving behavior might originate from agreements with DSOs regarding the maximum power feed-in of the VPP. Starting from results without the cosimulation framework, we successively include the cosimulation environment and domain-specific simulators to demonstrate that the results do not change under the assumption of an ideal communication network in the same scenario. Eventually, a slightly different scenario is presented where the communication network is supposed to impair the control loop between the power system and the market due to communication device failures, thereby, showing an exemplary use case for the cosimulation architecture presented in this paper.

6.1. Simulation Scenarios and Models. The investigated power system is a part of the IEEE European Low Voltage Test Feeder. The loads in the test feeder are replaced with buildings, each incorporating a PQ-load. Furthermore, several of these buildings feature stationary batteries and solar generation. The battery storages are controlled by a peak-shaving algorithm which is implemented in the market simulator. The peak-shaving algorithm is supposed to represent a VPP operator that utilizes its aggregated storage in a grid supporting way. That is why in a real world application it needs to exchange measurement and control values with the battery storages through the communication network.

In the first simulation scenario, the communication between batteries and the peak-shaving algorithm is ideal and not subject to any communication network failures. This scenario is used to verify the correct exchange of simulation data between the simulators involved in the cosimulation. For this reason, the simulation is first executed without cosimulation framework, connecting the peak-shaving algorithm directly to the power system simulation. Then, the cosimulation environment is introduced and afterwards the communication network simulator. The results of these three simulation cases are presented in Section 6.2.

The second simulation scenario, analyzed in Section 6.3, features a communication network failure in order to show a possible use case of the complete cosimulation environment.

The following equations are implemented in Modelica to simulate the power system components in the buildings. The PQ-loads in the buildings are modelled as ideal constant power loads, which means that the load current is directly dependent on the voltage at the grid connection point.

The implemented Modelica model of the solar generator determines the active power output P_{sg} by

$$\frac{P_{sg}}{3} = V_{oc} \cdot I_{sc} \cdot FF \cdot \frac{P_{sg,inst}}{P_0}, \quad (2)$$

under the assumption that the plant is operating at its maximum power point and where V_{oc} and I_{sc} are the open-circuit voltage and the short-circuit current, respectively. They vary with temperature and solar radiation, which can be modelled according to [27]. The term $P_{sg,inst}/P_0$ in (3) adapts the magnitude of the output power to the installed power of a specific solar generating unit, given that P_0 represents the installed power of the solar panel used in [27].

The model of the battery storages consists of a simple set of equations describing the derivative of the state of charge (SOC) as

$$\frac{d}{dt} \text{SOC} = \begin{cases} \frac{\eta_{ch} P_B}{C_B}, & P_B \geq 0, \\ \frac{P_B}{\eta_{disch} C_B}, & P_B < 0. \end{cases} \quad (3)$$

In addition, the battery model limits the SOC to be in the range between zero and one. The values specifying the battery capacity C_B , the charging efficiency η_{ch} , and the discharging efficiency η_{disch} are extracted from the extended CIM classes; see Section 5.1.

The VPP algorithm aims at stabilizing the voltage profile by reducing the power exchange between the grid and the buildings using a model predictive control approach. Therefore, the battery charging power P_B is set according to an optimal scheduling which is based on forecasts for solar radiation and load demand. To compensate for inaccuracies in the forecasts, the algorithm receives measurements of actual load demand, generated solar power, and battery state of charge from the power system simulator.

6.2. Comparison of Results with and without Cosimulation Environment. At first, both measurements and set points are neither passed through the communication network simulator nor the cosimulation framework. Instead, the control signals for the battery storages are directly supplied by the peak-shaving algorithm before each power system simulation step. Following this reference case, both simulators are connected through the cosimulation environment as described in Section 5.4. Still, the communication network simulator is not involved. The results depicted in Figure 8 present the voltage profile over time at one node in the test feeder and it can be seen that the results with and without the cosimulation environment are consistent. If the communication network had altered the exchanged data, control values, and measurements, the voltage profile would have changed. Next, we take this one step further and also introduce the communication network simulator. The communication network simulator acts as mediator between the power system and market simulators. Any data that is exchanged between the power system and market has to pass through the former. The communication network simulator is added to the cosimulation as presented in Sections 5.3 and 5.4. To verify that the three simulators are synchronized properly, we include an ideal communication network; that is, all messages are transmitted without any latencies. Evidently, an ideal communication should not affect the information exchange between power system and market, so that the

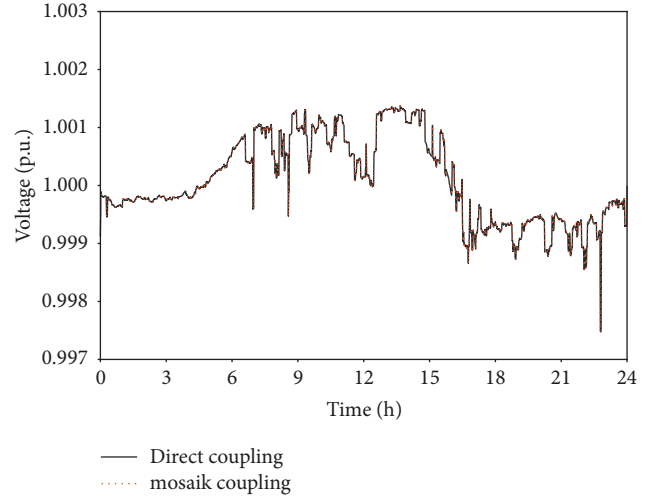


FIGURE 8: Comparison of simulation results of power system and market with and reference case without cosimulation environment.

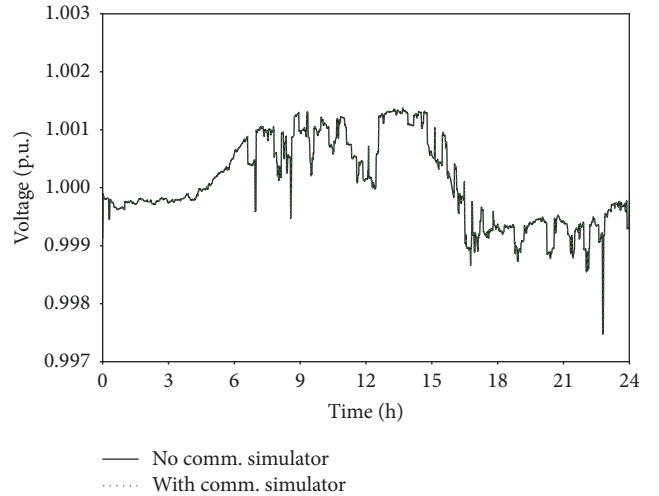


FIGURE 9: Comparison of simulation results with cosimulation environment and ideal communication network and reference case without cosimulation environment.

implemented scheduling for battery charging should perform equally.

In Figure 9, it is visible that the simulation results incorporating the ideal communication network do not deviate from the ones obtained without the communication network even though all messages are now passing through the communication network simulator. Thus, the results confirm a correct synchronization of the three simulators and the consistency of the implemented cosimulation environment.

6.3. Exemplary Cosimulation with Communication Network Failure. In this subsection, the three simulators are exchanging data in the same way as described at the end of Section 6.2. Only this time, the communication network simulator emulates a fault in the communication network which leads to a communication failure of one hour at the site of the

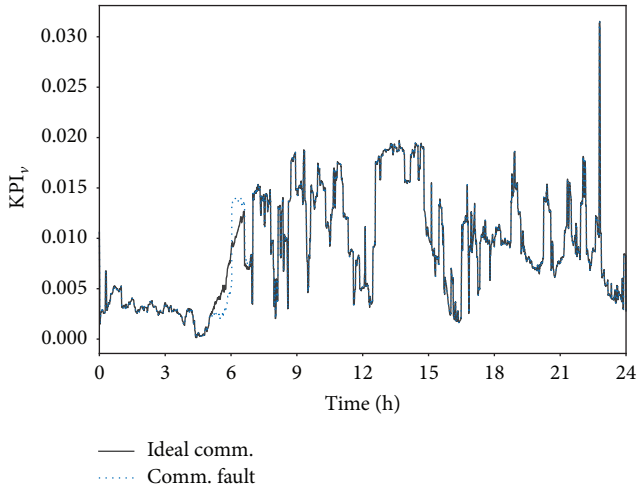


FIGURE 10: Comparison of voltage KPI for cosimulation with and without communication network fault.

VPP control algorithm. This represents slow phenomena case where the loop between power system and market simulator is affected.

To assess the impact of the communication fault on the whole network section, a voltage key performance indicator (KPI) as defined in (4) is applied to the results:

$$\text{KPI}_v = \frac{1}{N} \cdot \sum_{i \in \mathbb{N}} \left| \frac{v_i - 1}{\Delta v_{\max}} \right|. \quad (4)$$

Here, N is the number of considered voltage nodes, v_i denotes the voltage at one node in per unit, and Δv_{\max} is the maximum allowed voltage deviation. Thus, the KPI is representing the average absolute deviation at all network nodes in relation to the maximum allowed deviation. Figure 10 shows that a communication failure between hours five and six clearly affects the voltage profile of the power system. The application of the latest control values during the communication failure results in a charging of the batteries, while after the fault clearance the reactivated control discharges the batteries again. Hence, the entire battery capacity is being made available again for peak-shaving during the midday time. Due to the compensation behavior, the system returns back to a state with completely discharged batteries and consequently the same voltage profile as for ideal communication occurs some time after the fault. By modifying the communication network simulator input, it is also possible to emulate other faults such as faults at single buildings or package loss.

The applied market simulator focuses on the representation of VPP operators which manage as market participants their assets in a peak-shaving manner based on mathematical optimization. The operators scheduling may additionally take into account price trends at wholesale markets, for example, given as historical time series. An enhanced consideration of market dynamics can be obtained by an agent-based simulation of market participants and customers, for example, carried out with the Power TAC platform [11] integrated with the market simulator.

7. Conclusion

The contributions of this paper are a data model that includes three domains, power system, communication network, and market, and the architecture of a software environment to simulate multidomain scenarios for smart grids. The proposed data model facilitates the use of the software environment since the domain-specific smart grid component parameters and their interconnections can be modified and stored in a self-contained topology description. Due to our cosimulation approach we are able to take advantage of established domain-specific simulators for each domain. The proposed cosimulation environment covers use cases which are commonly investigated in literature and relate to power systems in conjunction with communication networks and use cases including the former two domains and the market. Simulation results of our cosimulation environment according to the proposed architecture confirm the consistency of the approach.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors gratefully acknowledge the financial support for this project by BMBF (German Federal Ministry of Education and Research) under Promotional Reference 03EK3567B.

References

- [1] W. Li, M. Ferdowsi, M. Stevic, A. Monti, and F. Ponci, "Cosimulation for smart grid communications," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2374–2384, 2014.
- [2] Z. Wang and Y. He, "Two-stage optimal demand response with battery energy storage systems," *IET Generation, Transmission & Distribution*, vol. 10, no. 5, pp. 1286–1293, 2016.
- [3] L. Exel, F. Felgner, and G. Frey, "Multi-domain modeling of distributed energy systems-The MOCES approach," in *Smart Grid Communications (SmartGridComm), IEEE International Conference*, pp. 774–779, USA, November 2015.
- [4] M. Uslar, M. Specht, S. Rohjans, J. Trefke, and J. M. Vasquez González, *The Common Information Model CIM: IEC 61968/61970 and 62325-A practical introduction to the CIM*, Springer Science & Business Media, 2012.
- [5] CEN-CENELEC-ETSI Smart Grid Coordination Group, "Smart grid reference architecture," (Accessed on 03.01.2018). [Online]. Available: https://ec.europa.eu/energy/sites/ener/files/documents/xpert_group1.reference.architecture.pdf, Nov. 2012.
- [6] E. Haq, D. Haller, K. A. Rahman, and B. Iverson, "Use of Common Information Model (CIM) in electricity market at California ISO," in *Power and Energy Society General Meeting, IEEE*, pp. 1–6, 2011.
- [7] J. Fremont, E. Lambert, C. Bouquet, O. Carre, D. Ilhat, and P. Metayer, "CIM extensions for ERDF information system projects," in *Power & Energy Society General Meeting, PES'09. IEEE*, July 2009.

- [8] K. Hopkinson, X. Wang, R. Giovanini, J. Thorp, K. Birman, and D. Coury, "EPOCHS: A platform for agent-based electric power and communication simulation built from commercial off-the-shelf components," *IEEE Transactions on Power Systems*, vol. 21, no. 2, pp. 548–558, 2006.
- [9] K. Zhu, M. Chenine, and L. Nordstrom, "ICT architecture impact on wide area monitoring and control systems' reliability," *IEEE Transactions on Power Delivery*, vol. 26, no. 4, pp. 2801–2808, 2011.
- [10] H. Lin, S. S. Veda, S. S. Shukla, L. Mili, and J. Thorp, "GECO: Global event-driven co-simulation framework for interconnected power system and communication network," *IEEE Transactions on Smart Grid*, vol. 3, no. 3, pp. 1444–1456, 2012.
- [11] W. Ketter, J. Collins, and P. Reddy, "Power TAC: A competitive economic simulation of the smart grid," [Online] Available: <http://linkinghub.elsevier.com/retrieve/pii/S0140988313000959>, vol. 39, pp. 262–270, 2013.
- [12] F. Schloegl, S. Rohjans, S. Lehnhoff, J. Velasquez, C. Steinbrink, and P. Palensky, "Towards a classification scheme for co-simulation approaches in energy systems," in *Smart Electric Distribution Systems and Technologies (EDST), International Symposium on IEEE*, pp. 516–521, September 2015.
- [13] C. Molitor, S. Gross, J. Zeitz, and A. Monti, "MESCOs-A multi-energy system cosimulator for city district energy systems," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2247–2256, 2014.
- [14] M. Mirz, L. Netze, and A. Monti, "A multi-level approach to power system Modelica models," in *Control and Modeling for Power Electronics (COMPEL), 2016 IEEE 17th Workshop on IEEE*, pp. 1–7, 2016.
- [15] K. Wehrle, M. Günes, J. Gross, and M. Günes, *Modeling and tools for network simulation*, Springer Science and Business Media, 2010.
- [16] W. E. Hart, J.-P. Watson, and D. L. Woodruff, *Pyomo-optimization modeling in python*, vol. 67, Springer, 2012.
- [17] B. P. Zeigler, H. Praehofer, and T. G. Kim, *Theory of modeling and simulation: integrating discrete event and continuous complex dynamic systems*, Academic press, 2000.
- [18] H. A. Tokel, G. Alirezaei, and R. Mathar, "Integrated network design for measurement and communication infrastructures in smart grids," in *Proceedings of the 26th International Telecommunication Networks and Applications Conference, ITNAC 2016*, pp. 258–264, Dunedin, New Zealand, December 2016.
- [19] L. Razik, M. Mirz, D. Knibbe, S. Lanke, and A. Monti, "Automated deserializer generation from CIM ontologies: CIM++ an easy-to-use and automated adaptable open-source library for object deserialization in C++ from documents based on user-specified UML models following the Common Information Model (CIM) standards for the energy sector," *Computer Science - Research and Development*, pp. 1–11, 2017.
- [20] S. Schütte, S. Scherfke, and M. Tröschel, "Mosaik: A framework for modular simulation of active components in Smart Grids," in *Proceedings of the IEEE 1st International Workshop on Smart Grid Modeling and Simulation, SGMS 2011*, pp. 55–60, October 2011.
- [21] S. Rohjans, E. Widl, W. Müller, S. Schütte, and S. Lehnhoff, "Co-Simulation of complex energy systems with mosaik and FMI," *At - Automatisierungstechnik*, vol. 62, no. 5, pp. 325–336, 2014.
- [22] S. Vogel, M. Mirz, L. Razik, and A. Monti, "An open solution for next-generation real-time power system simulation," in *Proceedings of the IEEE Conference on Energy Internet and Energy System Integration (EI2)*, pp. 1–6, Nov 2017.
- [23] M. Stevic, A. Estebarsari, S. Vogel et al., "Multi-site European framework for real-time co-simulation of power systems , IET Generation," *Transmission Distribution*, 2017.
- [24] (Accessed 2016-08-18) mosaik documentation –release 2.3.0. [Online]. Available: <https://media.readthedocs.org/pdf/mosaik/latest/mosaik.pdf>.
- [25] (Accessed 2016-08-18) Simpy documentation –release 3.0.9. [Online]. Available: <https://media.readthedocs.org/pdf/simpy/latest/simpy.pdf>.
- [26] S. Schütte, S. Scherfke, and M. Sonnenschein, "Mosaik - Smart grid simulation API: Toward a semantic based standard for interchanging smart grid simulations," in *Proceedings of SMARTGREENS*, pp. 14–24, April 2012.
- [27] W. Zhou, H. Yang, and Z. Fang, "A novel model for photovoltaic array performance prediction," *Applied Energy*, vol. 84, no. 12, pp. 1187–1198, 2007.




Hindawi

Submit your manuscripts at
www.hindawi.com

