MESCOS—A Multienergy System Cosimulator for City District Energy Systems

Christoph Molitor, *Graduate Student Member, IEEE*, Stephan Groß, Jakob Zeitz, *Student Member, IEEE*, and Antonello Monti, *Senior Member, IEEE*

***Abstract*—This work introduces a multidomain simulation platform that enables a holistic analysis of city district scale energy systems. The objective for the development of the simulation platform is to provide a tool that supports the design of control and energy management algorithms for those systems. The platform allows long-term simulations of a large number of buildings, including internal energy supply or energy conversion systems, in combination with external energy supply systems like the electrical grid. The simulation of those physical systems represents the environment for sophisticated control and energy management algorithms that can be tested on the platform. The concept of this work is to combine commercial-off-the-shelf software packages, here simulators and runtime infrastructure (RTI), to a high performance multidomain cosimulation platform. The high performance of the platform regarding computation time has been achieved by exploiting the parallel computing capabilities of modern simulation servers. Especially the computation time of large numbers of instances of Modelica-based models has been reduced significantly by the development of the parallel execution framework (PEF). The implementation of the PEF, including the interface to the individual models and to the RTI, is described in detail. The partitioning of the simulated system among different simulators does not influence the simulation results, as shown on the basis of a small-scale simulation scenario. The performance regarding the computation time is demonstrated on several example simulation scenarios showing the scalability of the platform.**

***Index Terms*—Building energy system (BES), cosimulation platform, multienergy system, smart grid, urban energy system.**

I. INTRODUCTION

THE TRANSITION of residential buildings from passive energy consumers to active players in the energy supply system is seen as an important step on the way to the smart grid. As residential buildings are responsible for 27% of the primary energy consumption [1], where 68% is used for space heating [2], there is a significant potential for energy savings but also for demand side management (DSM) and demand response (DR) [3]. According to [4], electrothermal heating systems like heat pumps (HP), electric heaters (EH), and combined heat and power (CHP) systems can contribute substantially to integrate renewable energy sources (RES) into the power network. This can be achieved by exploiting the operational flexibility of the heating and cooling systems, thanks to thermal storage, which is very cost-effective and can be provided by water tanks or even by using the building mass itself as short-term storage. Thus, so called building energy systems (BES) [5] comprising electrothermal heating systems and storage systems can contribute to an efficient integration of RES into the energy supply systems.

The assessment of the true potential and capability of a city district energy system to support the integration of RES requires the holistic analysis of the multidomain system comprising the energy supply infrastructure, a large number of heterogeneous BESs, as well as control and energy management systems (EMSs) [6]. The inclusion of the energy supply infrastructure into the holistic analysis provides insight in effects, such as line congestion in the electrical power network, that may not be investigated in individual buildings but that depend on the collective behavior of a group of buildings. The analysis of a residential district, with the goal to exploit the synergies between different kinds of BESs, involves also the control systems at different levels. The motivation for the development of the simulation platform presented here is to test and analyze the performance of novel energy management and control concepts on an urban district scale. Here, the focus regarding controllable devices is on residential electrothermal heating systems. However, the platform also allows the analysis and simulation of city district energy systems that are not strictly related to buildings, e.g., the integration of electric vehicles into the grid.

The remainder of this paper is organized as follows. Section II introduces the requirements for a simulator of city district scale energy systems. Section III provides a literature review followed by a detailed description of the approach of the platform presented in Section IV. Section V introduces the implementation of the platform, and Section VI the platform integration. Section VII presents sample simulation results and their interpretation in terms of satisfaction of the requirements.

II. REQUIREMENTS FOR CITY DISTRICT ENERGY SYSTEM SIMULATION PLATFORMS

The development of the presented simulation platform has the objective to provide a tool that enables the holistic analysis of city district energy systems. In this section, first the requirements for the simulation platform regarding the system

---

Manuscript received October 04, 2013; revised February 06, 2014 and April 02, 2014; accepted May 13, 2014. Date of publication June 30, 2014; date of current version November 04, 2014. Paper no. TII-13-0768.

The authors are with the Institute for Automation of Complex Power Systems, E.ON Energy Research Center, Rheinisch-Westfälische Technische Hochschule (RWTH) Aachen University, 52074 Aachen, Germany (e-mail: cmolitor@eonerc.rwth-aachen.de; sgross@eonerc.rwth-aachen.de; jzeitz@eonerc.rwth-aachen.de; amonti@eonerc.rwth-aachen.de).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2014.2334058

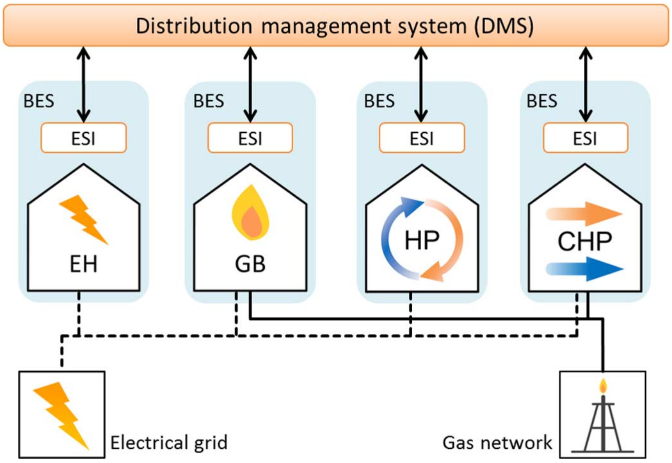

Fig. 1. Schematic representation of a possible future city district energy system.

modeling are introduced on the basis of a possible scenario for a future district energy system. The required modeling features induce further requirements regarding the computational performance of the simulation platform, introduced later in this section.

The scenario as illustrated in Fig. 1 comprises a significant number of heterogeneous BESs, differing regarding the internal heating system. Some buildings are refurbished and equipped with a modern heating system, like CHP or HP. Others are nonrefurbished and equipped with a heating system like a classical gas boiler (GB) or EH. Furthermore, in addition to Fig. 1, some of the BESs comprise a photovoltaic (PV) system. In general, it is assumed here that the buildings are equipped with an energy service interface (ESI), connecting the building and its devices with a distribution management system (DMS) as part of the future smart grid [7]. Those ESIs control the operation of heating or cooling systems in the buildings as well as other dispatchable devices like washing machines or dish washers. The objective of these control actions is to guarantee the user comfort, while saving energy or reducing the energy bill. This could be achieved, e.g., by shifting the operation to time periods with low electricity prices or with large power output from own PV or CHP systems. Furthermore, the user may accept a lower indoor temperature during times of absence.

In the future, external control signals like the mentioned electricity price, might be feasible, e.g., in the case of an aggregator that has to follow certain load curves or that offers services to the grid operator to support a stable operation of the local grid [8]. Thus, the aggregator as operator of the DMS has to perform day-ahead planning of the available resources but also has to react to short-term deviations from the anticipated behavior. In order to enable the simulation of the scenario described above, the simulation platform has to fulfill the requirements introduced in the following sections.

A. *Requirement (Req.) 1: Dynamic Simulation of BES*

The platform has to provide dynamic simulations of complete BESs also incorporating the influence of the inhabitants on the behavior of the BESs [9]. In more detail, it is required that the platform has to provide the possibility to simulate dynamic building models, as the actual indoor temperature and its variations are important input for control algorithms. For example, control algorithms exploiting the allowed indoor temperature range for DSM, depend among others on the actual indoor temperature, and the thermal behavior of the building. Furthermore, the inhabitant and its energy-related behavior has to be represented as far as it acts as a disturbance for the control algorithms. This element allows the evaluation of the capability of control algorithms to react to disturbances like the use of an oven where the exact time of use cannot be predicted. Furthermore, the consumption of electric energy by a resident leads to waste heat, which contributes to the supply of the thermal energy demand and thus reduces the load for the heating system. Also, a dynamic representation of the heating system of the building is necessary, as it imposes constraints on the possible control algorithm. Thus, besides nominal power, the startup behavior and forced shutoff periods as well as the dynamic efficiency of the specific device also have to be modeled. For example, the efficiency of a HP strongly depends not only on the outdoor temperature but also on the flow and return temperature, which are directly coupled with the state-of-charge (SoC) of the thermal storage [10]. Therefore, also the thermal storage has to be modeled in detail to represent a realistic behavior.

B. *Req. 2: Simulation of Energy Supply Infrastructure*

The simulation platform has to incorporate the simulation of the energy supply infrastructure in order to provide information about the state of the networks to the control systems. This feature is required in order to investigate control actions, which do not only depend on internal conditions of the BES but also on external conditions, like the state of the energy supply systems [11]. In the case of the electrical network, this information could be the voltage at certain nodes but also information about grid congestions and overload of grid components, e.g., like the load at the local transformer. Regarding the purpose of the platform and considering that the control actions on electrothermal heating systems or other dispatchable devices have a response time in the range of minutes, steady state power flow analyses are, in general, sufficient to provide the required information.

C. *Req. 3: Simulation of Control and Energy Management Algorithms*

Control and energy management algorithms at different system levels are critical for the efficient operation of a city district energy system [12]. Thus, it is required that the simulation platform provides the possibility to represent those algorithms at the application control level as well as at the system control level. Referring to the example scenario, the DMS executing the energy management algorithms would represent the system control level. Reference values or commands determined at this level would be passed on to the application control level for execution. The application control algorithms turn the received signals into actual commands considering the current state and operational boundaries of the system. That means, e.g., the application control can overrule the system control commands in case of violations of the operation limits. The platform should provide sufficient flexibility to implement different control architectures and philosophies, e.g., centralized control algorithms versus decentralized control algorithms implemented as multiagent systems (MASs). This requirement forces the platform to provide data and information exchange between each controller within its own control simulation platform and the models of the physical component.

D. Req. 4: Computational Performance

The simulation of a city district energy system carries requirements also regarding the computational performance, in addition to the mentioned requirements regarding modeling and simulation. As a city district like the one described in the example scenario comprises at least hundreds of individual BES, the platform has to be capable to integrate and execute a large number of dynamic models of BES. Additionally, it is required to simulate time periods of several weeks up to a full year, in order to assess the performance of energy management algorithms for city district energy systems. The simulation of longer time periods allows the evaluation of the algorithm under a variety of different operating conditions and thus will lead to more meaningful results given that the scenario does not depend on specific conditions of one or several days. In order to simulate and compare different scenarios or control strategies, the simulation platform should be able to execute those long-term simulations within a reasonable time.

III. BACKGROUND AND LITERATURE REVIEW

In literature, several tools have been presented for simulating city district or urban energy systems. The review in [13] provides an overview of existing simulation and optimization tools for city district energy systems. Especially, the tools HOMER [14], EnergyPlan [15], and DER-CAM [16] are used for running simulations of urban energy systems. HOMER supports the design of microgrid systems by evaluating different configurations based on their life-cycle cost. Furthermore, HOMER allows the evaluation of the technical feasibility of the system. The analyses conducted by HOMER are based on simulations of a full year of operation; however, the simulations are based on hourly input data and only rule-based control strategies can be implemented [14]. Similarly, the tool EnergyPlan [15] allows the execution of technical analyses as well as market analyses at system level [13]. EnergyPlan also uses hourly input data for the simulations. The third tool, DER-CAM, is a technoeconomic optimization model, which provides as result, e.g., the lowest-cost configuration of distributed generation technologies for a specific building. However, it is also possible to optimize the configuration of microgrids [13]. In DER-CAM, the resolution of the input data can be up to 5 min [16]. However, all the tools above have limitations that make them unfit for the scenario outlined in Section II. First of all, these tools represent buildings in terms of their thermal demand profile; thus, studies exploiting the flexibility of the indoor temperature are not possible. Also, all these tools do not include the simulation of the energy supply infrastructure, e.g., of the electrical grid. Furthermore, they do not provide the option to integrate sophisticated control algorithms. Especially, it is not possible to integrate different control objectives for the individual BES or distributed control and energy management algorithms.

The IDEAS tool presented in [10] and [11] overcomes some of the mentioned shortcomings by using Modelica to build the model of a neighborhood of 33 residential buildings, including dynamic models of buildings, and also a model of the electrical grid suitable for power flow analysis. In this work, several rule-based DSM algorithms have been implemented in Modelica and compared to each other. According to [10], simulating the neighborhood for 1 year took between 1.2 and 4.3 days. Due to the complete implementation into one Modelica model, it can be expected that the computation time would increase significantly for larger scenarios. Furthermore, the integration of more sophisticated energy management algorithms including forecasting and planning would require incorporating other simulation tools.

In contrast to IDEAS that implements the models of all system components in one simulation tool, other approaches couple different commercial-of-the-shelf (COTS) simulators into one simulation platform. In order to achieve this integration, a runtime infrastructure (RTI) is necessary to take care of the time management and the data distribution management. This approach is very common in order to simulate the interaction of power system and communication systems as described in [17]–[19]. This approach reduces significantly the implementation and modeling effort, increases the reliability, due to the usage of established simulation packages, and reusability of models and simulation schemes.

BCVTB [20] and MOSAIK [21], [22] apply the concept of coupling different simulators for multidomain city district simulations. BCVTB demonstrates the coupling of EnergyPlus [23] and Dymola [24] as Modelica environment. According to the authors, the long-lasting computations could be overcome by running instances of EnergyPlus in parallel, which has not been demonstrated. MOSAIK describes another interesting approach for the simulation of active components in smart grids. The MOSAIK framework proposes an RTI that allows for automatically composing simulation scenarios, randomly or following defined patterns, out of existing models. Therefore, the models including the simulator have to be wrapped in several interface layers. According to the developers, the focus has been on the semantic describing the models and their interfaces and not on the simulation performance, which is instead a main goal of this work. Another interesting RTI for cosimulation, the simulation message bus (SMB), is presented in [25] and [26]. The SMB is a powerful RTI allowing connected clients to exchange data using various protocols. Thus, the SMB can be used during the whole development process of a controller, e.g., even when the simulation of the system model is later replaced by a physical system.

The simulation framework GridSpice exploits similar to this work the possibility to parallelize certain simulations in order to decrease the computation time [27]. The focus of GridSpice

TABLE I
COMPARISON OF REVIEWED CITY DISTRICT ENERGY SYSTEM SIMULATION TOOLS/PLATFORMS

Req.	HOMER	EnergyPlan	DER-CAM	IDEAS	BCVTB	MOSAIK	GridSpice
1				X	X	X	
2				X		X	X
3				(X)	(X)	(X)	
4	X	X	X				X

is on the simulation of transmission grids and distribution grids in combination with market models. GridSpice has not been used for simulating a granularity down to the individual building.

In Table I, the review of the simulation tools and platforms is summarized with respect to the requirements of Section II. The SMB is not listed here, as it is functional to the RTI rather the complete simulation platform.

MESCOS aims at overcoming the shortcomings of these simulation tools and fulfilling all the requirements. The design and implementation of MESCOS focus on large-scale simulations and computational performance, while using as much as possible COTS software packages; thus, being highly suitable for professional applications.

IV. PROPOSED APPROACH TO THE SIMULATION PLATFORM

The presented simulation platform combines two approaches: 1) integrating existing COTS software tools and libraries; and 2) exploiting parallel computing features in order to speed up the simulations. Composing the simulation platform of COTS software packages allows to use the advanced features of commercial tools as well as the support, which is important for professional applications. The drawback of COTS software tools in the relevant domains is that parallel computing features are still not exploited very well. Nevertheless, existing experience with certain software tools as well as existing libraries can be utilized.

Here, the system to simulate as described in Section II has been split in three different layers: 1) the entity layer; 2) the network layer; and 3) the system control layer. The individual entities, here BESs, are simulated within the entity layer, the energy supply infrastructure in the network layer, and the DMS within the system control layer. This specific partitioning has been applied due to the wide availability of domain-specific simulators for the involved domains. This partitioning of the system imposes two challenges. The first one is that the system has to be split in several parts and the partitioning has to preserve the relevant interactions between the parts. The second challenge is that the data, as well as the time, must be synchronized among the simulators. Thus, an appropriate time and data distribution management is required and provided by an RTI.

In order to decrease the computation time, the execution of the simulation has been parallelized in this work at two different levels.

1) The execution of the individual layers can be parallelized.
2) Parallel execution has been applied within the individual layers, here mainly the entity layer.

In general, different simulation layers of the platform can be executed in parallel or in series. The choice of the execution mode for appropriate data synchronization depends on the chosen calculation method within the individual layers. In the case of coupled simulators, each running a continuous time simulation of a subsystem, the simulations in each tool are executed in parallel and the values of the coupling variables are exchanged at a defined simulation time, called macro-time step (MTS). Internally, the simulators might use different time steps, called micro-time step (μTS). In case of highly dynamic coupling variables, the MTS would be chosen close to the μTS. In the case of less dynamic coupling variables, the MTS could be chosen as a larger integer multiple of the μTS, thus reducing the synchronization rate.

In the case of a continuous time simulation of a subsystem in one simulator and a discrete time analysis in another simulator, a serial execution of the simulators might be favorable. For example, in the case of a BES and an EMS, the EMS is called at defined time intervals to determine new operation schedules based on the current state of the BES. Thus, the continuous time simulation of the BES is executed for the defined time interval and then paused while the EMS performs its calculations. After the EMS completes the scheduling, the simulation of the BES is continued incorporating the new schedules.

The synchronization of the presented simulation platform can be configured in order to support the parallel and the serial execution of individual layers. Furthermore, a hybrid operation mode is possible where, e.g., two continuous time simulators run in parallel and a third discrete time simulator performs a snapshot analysis at defined time intervals.

Regarding the parallelization within the individual layers, the main focus in this work is on the parallelization of the simulation within the entity layer. As seen by the literature review, Modelica as a multidomain modeling language is popular in the field of building and BES simulations. The availability of commercial and open source Modelica tools as well as libraries for BES simulations supports the use of Modelica for professional and academic purposes. However, all available Modelica modeling and simulation environments and solvers are single core applications. Thus, the parallel execution of Modelica models within those environments can only be achieved by running several instances of those that does not offer a comfortable simulation setup and might lead also to license issues. However, some Modelica environments offer the option to export the Modelica-based models as C-Code that can be compiled and executed as stand-alone application or integrated into other applications. Thus, the approach for this work is to develop the parallel execution framework (PEF) that is able to execute large number of instances of Modelica models, which have been exported as C-Code. This allows using the comfortable and powerful modeling features of a Modelica environment, while enabling a fast computation time of a large number of instances. Furthermore, the PEF provides the interface between the individual instances and the RTI and allows a fully automatic setup of all instances and of the data and time synchronization.

The energy supply networks that connect the individual entities are simulated within the network layer of the simulation platform. In the case of city district energy systems, the most relevant networks are gas, electricity, and district heating. Each of the networks can be simulated within a specific simulator that can run in parallel. As commercial simulators are often still single core applications, the execution of each specific simulator will be restricted by the computation power of a single CPU core.

The function of the system control layer is to integrate higher level control, e.g., energy management algorithms. Depending on the integrated tool, the implemented energy management algorithms can be solved in parallel. For example, MATLAB from Mathworks [28] as a popular platform for fast prototyping of control algorithms provides a special toolbox for parallel computing, which can lead to a significant speedup. Furthermore, frameworks for implementing MASs, which are suitable for EMSs are naturally multicore applications.

V. IMPLEMENTATION

Following the described approach in the previous section, different simulators have to be interfaced and coupled through an RTI. In this work, the commercial software package TLK Inter Software Connector (TISC) suite by TLK Thermo [29] has been used which offers interfaces for several simulation tools. Furthermore, it provides a C/C++ Application Programming Interface (API), which can be used to integrate tools that are not primarily supported. The interfaces and the C/C++ library establish a connection via transmission control protocol / Internet Protocol (TCP/IP) to the TISC server. Thus, the overall simulation can be distributed to multiple computers.

The general procedure for the data and time synchronization between the simulators through the TISC suite can be described as follows. The simulators connect to the TISC server and register the output and input variables. During the simulation, the simulators will provide updated values for the output variables, which serve as input variables for the other simulators. Vice versa, each simulator expects updated values for the registered input variables. Variables can be scalars, vectors, or matrices of type integer, double, or string. During the simulation, the simulators send the values of the registered variables to the server after simulating one MTS. Then, they wait for a synchronization signal from the server that indicates that the other simulators also finished the simulation of the MTS. After receiving the synchronization signal, the simulators request the values of the input variables for their simulation from the server and start the simulation of the next MTS.

In the following sections, different simulators used in this work within the layers are described as well as the implementation of the interfaces to the TISC suite.

A. Network Layer

For the simulation of the energy supply networks, the power system simulator Neplan by BCP [30] has been integrated into the simulation platform. Neplan provides the capability of simulating besides the electrical network, also gas and district heating networks. Here, the network states are determined by executing Neplan's load flow analysis, which provides steady state results for the actual load conditions of the network. Neplan can execute custom code compiled as a dynamic link library (DLL). This allows for integrating the interface to the TISC server into the simulator.

During a simulation step, the DLL requests the associated data sent by the other simulators from the TISC server and assigns it to the representing load in the grid model. After executing a load flow calculation, the values of the defined variables, e.g., voltage at certain nodes, are transmitted to the TISC server.

B. System Control Layer

Within the system control layer, MATLAB/Simulink in combination with the International Business Machines Corporation (IBM) ILOG Optimization Studio [31] has been used for the implementation of energy management algorithms. MATLAB/Simulink is widely known and accepted as a tool for fast prototyping. However, also other tools may be integrated in its place depending on the expertise, legacy, and also depending on the structure of the control algorithms.

Here, the used sample energy management algorithm has been integrated into the simulation platform via the Simulink interface provided by the TISC suite. This control algorithm is mainly a scheduler to determine the operation schedule of the electrothermal heating systems at certain time intervals, here every 24 h. At each synchronization time step, the control signals are sent to the entity layer, in order to actuate the calculated schedules. The scheduling itself is done by solving a mixed-integer program within the IBM ILOG Optimization Studio called from MATLAB/Simulink.

C. Entity Layer

The simulation of the BES models within the entity layer is handled by the PEF. The PEF provides the interface between the TISC suite and the entity models and manages the efficient and parallelized execution of the entity models. The entity models themselves have been developed within the Modelica environment SimulationX by ITI [32] in combination with the GreenBuilding library [33], [34]. SimulationX has a comfortable C-Code export, which allows a straightforward integration of the model code into other projects.

The objective for the development of the PEF is to enable an automatic setup of highly parallelized simulations and the required interfaces for data exchange. Thus, the PEF first reads a file containing configuration data of the platform like the MTS, the simulation end time, as well as IP and port of the TISC server. A second input file describes the specific simulation scenario by listing all entities, here BESs, including unique identifier, model type, number of inputs and outputs, parameters, as well as the specific μTS. The PEF

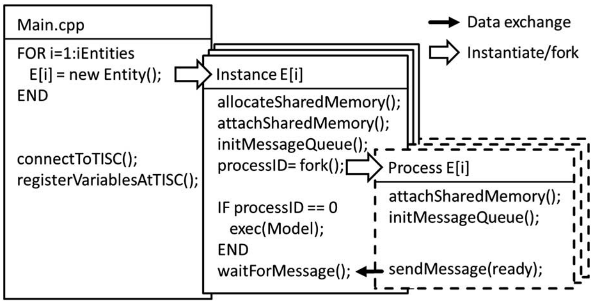

Fig. 2. Initialization procedure of the PEF.

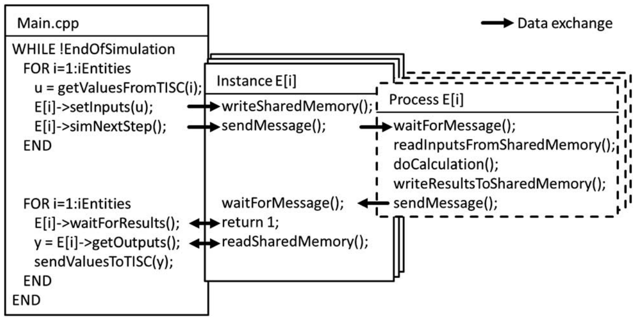

Fig. 3. Data exchange procedure within the PEF during runtime of the simulation.

creates separate simulation processes for all entities, which can be executed in parallel. Inter-process communication (IPC) mechanisms are used for establishing a communication channel between main process and the entity processes for data and time synchronization. More specifically, the data exchange is realized with shared memory, whereas the time synchronization is done with the help of the operating system's message queues. For the data exchange with other simulators, the main process establishes a TCP/IP connection to the TISC server.

Each entity process consists of the source code of a specific model type, the solver exported from SimulationX and a wrapper, which contains functions to control the model execution and to exchange data with the main process. The integration of new Modelica-based models is as simple as copying the source code files of the wrapper into the folder of the model source code and execute the compile script.

During the initialization procedure as shown in Fig. 2, the PEF creates an instance for each entity encapsulating the entity parameters provided by the scenario description file. Within the instance, shared memory is allocated and attached. Furthermore, a message queue is initialized for receiving messages from the later simulation process. Then a new process is created by forking (*fork()*) the current one. The new process executes (*exec(Model)*) the entity model based on Modelica code, attaches the shared memory, and connects to the message queue to communicate with the entity instance in the main process.

The interaction between the main process, the entity instances and the entity processes during the runtime of the simulation is described in Fig. 3. The *while*-loop in the main process is executed until the end of the simulation. In the first section of the loop, the input values are requested from the TISC server. Then the values are handed over to the entity instance by calling a method which writes the values to the shared memory. Calling the entity method *simNextStep()* by the main process, triggers the entity instance to send a message to the entity process. Upon receipt, the entity process reads the input values for the entity model from the shared memory and starts the calculation. After finishing the calculation, the entity process writes the result values to the shared memory and informs the entity instance that the simulation step is finished via message. Within the second section of the *while*-loop in the main process, the main process waits until all entities have finished their calculation. After the calculations are finished, the entity instance reads the results from the shared memory, which are returned to the main process. The main process sends the results to the TISC server and the next simulation step can start.

D. Hardware and Operating System Configuration

The PEF is optimized to run on a multicore computer system. The framework uses the IPC mechanisms, which are defined in the Portable Operating System Interface (POSIX) standard [35]. The POSIX standard is defined in the IEEE 1003 [36] and is used by many UNIX-based operating systems (OS). For the demonstration in this work, the simulation of the entity layer is executed on a 32-core Linux server with a clock speed of 2.1 GHz.

The TISC server and the other simulation layers, system control and network layer, run on a server with 16 cores using Windows Server 2012 OS. In order to achieve a fast simulation of the single core tool Neplan, CPUs with a high clock frequency of up to 3.8 GHz have been chosen.

VI. PLATFORM INTEGRATION

This section gives a short description of the used BES models as well as the data flow between different simulators. Here, the focus is not on the verification of the models themselves as they can be replaced by verified models but on the verification of the integration into the simulation platform.

A. BES Models

One building model representing a single family house has been implemented. The building has been modeled as one single thermal zone. Based on this building model, four BES models have been developed differing regarding the used heating system. One is equipped with a GB and three are equipped with electrothermal heating system like EH, CHP, or HP. Every BES comprises a thermal storage whose volume can be defined by a parameter. The heat delivery system as subsystem of the heating system, which supplies the building with thermal power, is modeled as a floor heating system. This subsystem comprises several valves, a circulation pump, and volume flow controllers. Furthermore, all BES models contain a PV system that can be deactivated by parameter. The internal controller of each BES processes measurements like the temperature within the water tank, the return and

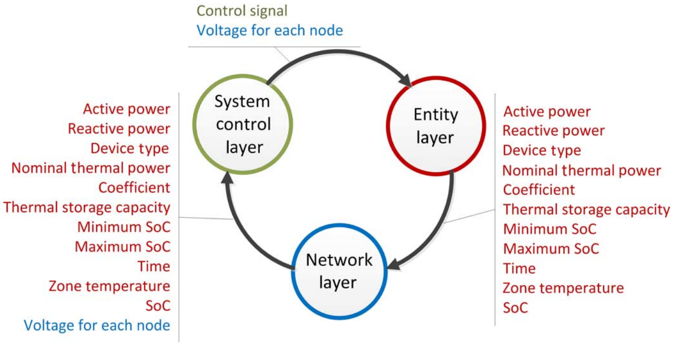

Fig. 4. Data flow between the layers of the simulation platform including the exchanged variables.

flow temperature and the outdoor temperature to generate the control signals for the heater. In the case of electrothermal heating systems, the controller considers also external control signals which might be provided by the DMS via the ESI. In order to guarantee user comfort, the controller can overrule the external control signal. For example, in the case of a very low SoC, the controller turns ON the heating system until a preset SoC is reached again. In the dual way, excessively high temperatures in the water tank are avoided.

The outdoor environment is represented by a weather model providing solar irradiation, temperature, and wind speed. These data are used to calculate the energy transfer between the thermal zone and the environment, by considering not only the temperature difference but also the solar irradiation on each individual boundary. The weather data also serve as input for each PV system, consisting of panels and inverter. The weather data are representative for a city located in the area of the Ruhrgebiet, Germany, and have been provided by DWD [37]. The modeling of the BESs has been done with SimulationX and the GreenBuilding library.

B. Data Exchange Between Layers of Simulation Platform

In the following examples, different simulators have been executed in series. The individual variables passed from one platform layer to another are described in Fig. 4.

First, the BES models within the entity layer are executed for one MTS. The output vector of each BES model contains dynamic variables representing the current state as well as parameters (e.g., device type) describing the system and are required by the sample algorithm within the system control layer. The output variables are sent via TISC server to the network layer, which use the active and reactive power values of each entity to update the according values in the grid model and execute a load flow calculation to determine the node voltages. The system control layer then determines a control signal for each BES. The control signals as well as the node voltages at each BES are the input variables for the next simulation step of the entity layer.

C. BES Model Integration

As described in Section IV, the models of the BESs have to be integrated into the PEF of the entity layer. Therefore, the models have to be exported to C-Code using the export features of SimulationX. During the export process, inputs, outputs, and parameters, which should be later accessible, have to be defined. Furthermore, the solver to be exported together with the model must be selected. Here, the ITI-fixed time step solver provided by SimulationX has been chosen. After exporting the model with the solver and adding the interface wrapper of the PEF, the code can be compiled.

In order to guarantee a stable but also efficient execution, an appropriate μTS has to be chosen. While a smaller μTS yields better accuracy, it increases the computation time. In order to determine a μTS value that realizes a tradeoff between accuracy and simulation speed, a numerical sensitivity analysis has been performed. To this aim, the exported models have been compiled and simulated with different μTS settings of the solver. The largest μTS resulting in a maximum deviation of 0.5% compared to a simulation with a μTS of 10 ms has been chosen. For the models used for this work, the μTS has been between 100 and 400 ms.

D. Platform Integration

In order to verify the integration of different layers within the platform, a simplified scenario has been used. 1) The scenario has been completely simulated within SimulationX, thus only using Modelica components and libraries. 2) The scenario has been simulated within the presented simulation platform using MTSs of 3 s and of 60 s. The scenario consists of a feeder supplying three BESs. The first and third BESs have an EH, whereas the second BES has a GB as heating system. The EH within the first BES is turned ON after 120 s and the one in the third BES after 60 s.

Fig. 5 shows the node voltage as well as the SoC and the electrical power consumption of the third BES. The voltage drops the first time after 60 s as the EH of the third BES is turned ON. The second voltage drop after 120 s occurs due to the turning-ON of the EH in the first BES connected to the same feeder. Furthermore, the power consumption of the third BES as well as the SoC increases after 60 s when the EH is turned ON. The simulation results from the Modelica simulation as well as from MESCOS are well aligned, which verifies the proper time and data synchronization. However, due to the system partitioning in this work, the variable values are only available for the other simulators after synchronization. Thus, certain dynamics are invisible for the other simulators and the MTS has to be chosen according to the requirements.

VII. ANALYSIS OF CONFORMANCE TO REQUIREMENTS

This section presents some sample simulation results and analyses the results regarding the conformance to the requirements as introduced in Section II. The analysis of Reqs. 1, 2, and 3 is carried out qualitatively showing that the simulators provide the required variables and the simulator coupling emulates the correct interaction. The analysis of Req. 4, regarding the computational performance, provides detailed computation times for different scenarios. The simulation results presented in the following sections are based on a scenario consisting

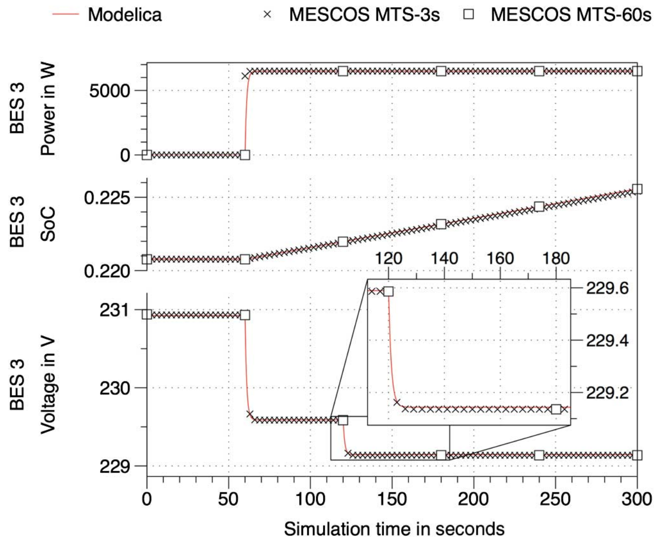

Fig. 5. Node voltage, SoC, and active power of a BES simulated in SimulationX and in MESCOS with a MTS of 3 and 60 s.

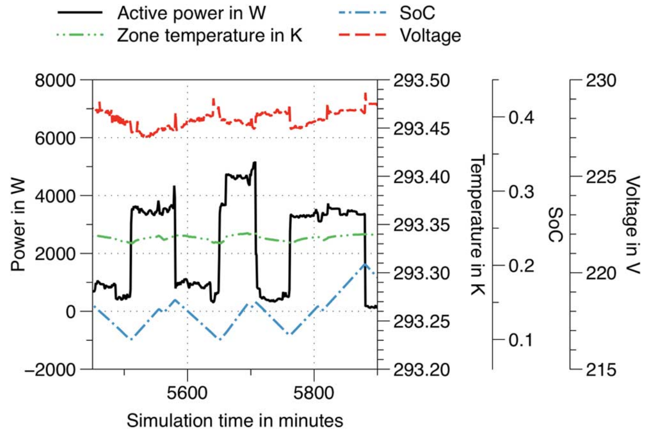

Fig. 6. Active power consumption, indoor temperature, and SoC of a sample BES with HP as well as the voltage at the point of common coupling.

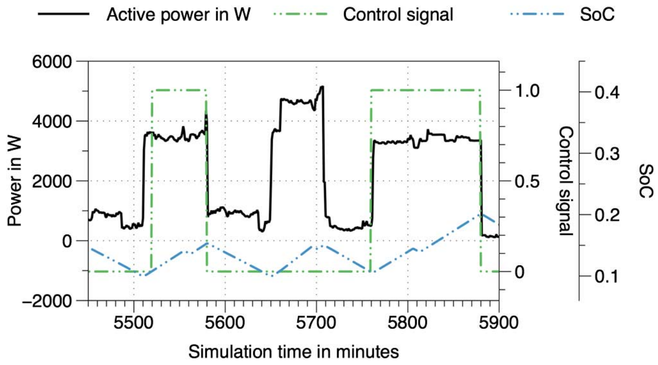

Fig. 7. Control signal, power consumption, and SoC of BES with HP.

of 146 BES differing in heating system, storage size, and PV system. The model of the electrical network has been implemented according to the suburban grid model presented in [38]. The radial network consists of a transformer, with rated power of 630 kVA, supplying 10 branches of different lengths. Each branch supplies different number of buildings, ranging from 4 to 32. For the analysis of the scalability of the simulation platform, a larger scenario consisting of 795 BES and a highly meshed distribution network based on a real grid data has been used. The MTS has been set to 60 s for the following simulations.

A. Req. 1: Dynamic BES Simulation

Fig. 6 shows the active power consumption, indoor temperature, SoC, as well as the phase to ground voltage at the house connection of a sample BES equipped with a HP for a time period of 450 min. It can be stated that the indoor temperature of the building stays constant with only small variations during the whole simulation period. This is in effect the objective of the control algorithm at the application control level. The operation times of the HP can be identified in the curve of the overall power consumption, where the times of high power consumption are mainly due to the operation of the HP. But also the operation of other appliances is represented. The trajectory of the SoC correlates with the operation of the HP. During times of HP operation, the SoC increases as the storage is charged, and during OFF-times, the storage is discharged to cover the thermal demand of the building. These results show that the platform performs a dynamic simulation of the BESs and variables needed for the higher level control and energy management algorithms are accessible. The algorithms can be tested in an environment with realistic boundary conditions, in support of the development of novel algorithms. In addition to the assessment of macroscopic system level effects of the algorithms, e.g., on the operation of the electrical grid (see Section VII-B), also the impact of the control algorithms within the individual BES can be analyzed. This microscopic view allows analyzing the behavior of the individual BES and the cause of deviations from the anticipated behavior.

B. Req. 2: Simulation of Energy Supply Infrastructure

Fig. 6 also depicts the trajectory of the grid voltage at the point of common coupling of the sample BES as result of the network simulation. The shown voltage profile results not only from the power consumption of the sample BES but also from the consumption of the BESs connected to the same branch. Nevertheless, the turning-ON and -OFF of the HP has an influence on the voltage profile as shown in Fig. 6. Although the voltage deviations are rather small in this case, the information would be available to use it as input for possible control algorithms. Also, information regarding the state of the local transformer or other grid elements would be accessible as input for control algorithms.

C. Req. 3: Integration of Control and Energy Management Algorithms

The requirements for the simulation platform regarding the integration of control algorithms focused on two levels: the application and the system control level.

The impact of the control at the application control level can be seen in Fig. 7, which shows the same simulation period as in Fig. 6, but instead of the indoor temperature and voltage curve, the control signal is passed from the system to the application

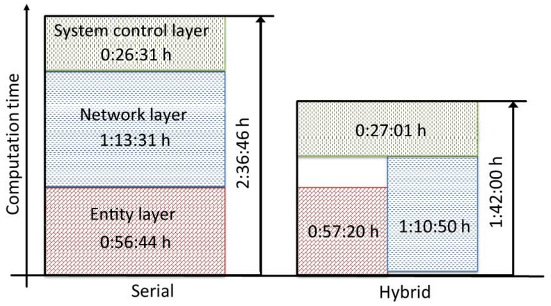

Fig. 8. Computation time for serial and hybrid execution of the individual layers.

control level. It can be noticed that the HP follows the external control signal well between 5750th and 5900th minute of the simulation. However, at around the 5650th minute, the HP turns ON although the control signal indicates OFF. This is because the SoC falls below a defined minimum value and the device controller representing the application control level overrules the external control signal. As soon as the SoC is above a defined limit again, the external control signal is active again and the HP turns OFF. These results demonstrate how control algorithms at application and system control level can be integrated and how they interact. These insights into the individual BES allow the analysis of the source of the deviations, e.g., from anticipated schedules and for improving the control and energy management algorithms. For example, learning algorithms could be applied to better estimate the coefficient of performance of the HP, which would lead to better scheduling results.

D. Req. 4: Computational Performance

First, the scenario with 146 BESs has been simulated while executing all the layers in series. The complete computation time as well as the computation time of the individual layers is given in Fig. 8. The sum of the individual computation times is equal to the overall computation time due to the serial execution. In order to speed up the simulation, the individual layers can be executed in parallel, taking into account that this comes with a delay in the data exchange between different layers. Nevertheless, for some simulations, this might be acceptable. For demonstration reasons, the entity layer and the network layer are executed in parallel. As expected and shown in Fig. 8, the computation time can be significantly reduced as the two layers are executed in parallel. However, the overall computation time is a little longer than just the sum of the longest computation time of the two parallel simulators and the computation time of the serial executed system control layer. This is due to the effect that two simulators request variables from the server at the same time, but they are served sequentially.

In order to evaluate the scalability of the simulation platform, a larger scenario has been simulated comprising the simulation of 795 BES and a model of a highly meshed low-voltage distribution grid. Again, the simulators are executed in series. The overall computation time for this scenario is 15:00 h. The computation time of the entity layer is 5:23 h of the network layer 6:27 h and of the system control layer 3:10 h. While the number of BES increases by the factor 5.45 from the small to the large scale scenario, the computation time increases by the factor 5.74. Thus, in the investigated range, the computation time of the simulation platform scales almost linearly.

VIII. CONCLUSION

This work introduces a cosimulation framework for city district energy systems mainly based on COTS simulators and software tools. Due to the use of an RTI, which provides interfaces for a variety of simulators as well as a $\mathrm{C/C{+}{+}}$ API, further simulators can be easily integrated into the simulation platform.

The concept of generating C-Code out of Modelica-based models enables the parallel execution of a large number of models. In order to achieve the parallelization of the model simulation, a custom PEF had to be developed. The choice of a process oriented approach and inter process communication based on the POSIX standard has proven to enable a stable operation and a simple integration of new models. For future works, further solver interfaces could be added in order to support variable time step solvers as well as C-Code from other Modelica environments.

The integration of a commercial power system simulator has been demonstrated, which could be done the same way for other simulators providing a $\mathrm{C/C{+}{+}}$ API. As the here used power system simulator is a single core application, like the most common ones, and does not exploit the computational power of multicore systems, future work should investigate how a partitioning of the electrical network simulation could decrease the computation time.

Furthermore, as the platform allows the observation of a large number of variables, a concept for the data management as well as for the visualization of the simulation results has to be developed.

REFERENCES

[1] Eurostat. (2012). *Europe in Figures* [Online]. Available: http://epp.eurostat.ec.europa.eu/

[2] European Environment Agency. (2012). *Household Energy Consumption by End-Use in the EU-27* [Online]. Available: http://www.eea.europa.eu/

[3] P. Palensky and D. Dietrich, "Demand side management: Demand response, intelligent energy systems, and smart loads," *IEEE Trans. Ind. Informat.*, vol. 7, no. 3, pp. 381–388, Aug. 2011.

[4] B. V. Mathiesen and H. Lund, "Comparative analyses of seven technologies to facilitate the integration of fluctuating renewable energy sources," *IET Renew. Power Gener.*, vol. 3, no. 2, p. 190, 2009.

[5] Z. Xu *et al.*, "Performance analysis and comparison on energy storage devices for smart building energy management," *IEEE Trans. Smart Grid*, vol. 3, no. 4, pp. 2136–2147, Dec. 2012.

[6] C. Molitor *et al.*, "New energy concepts and related information technologies: Dual demand side management," in *Proc. IEEE PES Innov. Smart Grid Technol. (ISGT)*, 2012, pp. 1–6.

[7] H. Farhangi, "The path of the smart grid," *IEEE Power Energy Mag.*, vol. 8, no. 1, pp. 18–28, Jan./Feb. 2010.

[8] L. Gkatzikis, I. Koutsopoulos, and T. Salonidis, "The role of aggregators in smart grid demand response markets," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 7, pp. 1247–1257, Jul. 2013.

[9] R. De Coninck, R. Baetens, D. Saelens, A. Woyte, and L. Helsen, "Rule-based demand-side management of domestic hot water production with heat pumps in zero energy neighbourhoods," *J. Build. Perform. Simul.*, vol. 7, no. 4, pp. 271–288, Jul. 2014.

[10] R. De Coninck *et al.*, "Modelling and simulation of a grid connected photovoltaic heat pump system with thermal energy storage using Modelica," in *Proc. 8th Int. Conf. Syst. Simul.*, 2010, 21pp.

[11] R. Baetens *et al.*, "Assessing electrical bottlenecks at feeder level for residential net zero-energy buildings by integrated system simulation," *Appl. Energy*, vol. 96, pp. 74–83, Aug. 2012.

[12] M. Bosman, "Planning in smart grids," dissertation, Univ. Twente, Twente, The Netherlands, 2012.

[13] M. Manfren, P. Caputo, and G. Costa, "Paradigm shift in urban energy systems through distributed generation: Methods and models," *Appl. Energy*, vol. 88, no. 4, pp. 1032–1048, 2011.

[14] T. Lambert, P. Gilman, and P. Lilienthal, "Micropower system modeling with HOMER," in *Integr. Altern. Sources Energy*, F. A. Farret and M. G. Simões, Eds. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2006, pp. 379–418.

[15] *EnergyPLAN|Advanced Energy Systems Analysis Computer Model* [Online]. Available: http://www.energyplan.eu/, accessed on Sep. 9, 2013.

[16] *Home|Microgrids* [Online]. Available: http://der.lbl.gov/, accessed on Sep. 9, 2013.

[17] X. Wang *et al.*, "Interfacing issues in multiagent simulation for smart grid applications," *IEEE Trans. Power Del.*, vol. 28, no. 3, pp. 1918–1927, Jul. 2013.

[18] K. Hopkinson *et al.*, "EPOCHS: A platform for agent-based electric power and communication simulation built from commercial off-the-shelf components," *IEEE Trans. Power Syst.*, vol. 21, no. 2, pp. 548–558, May 2006.

[19] 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 Trans. Smart Grid*, vol. 3, no. 3, pp. 1444–1456, Sep. 2012.

[20] J. Huber and C. Nytsch-Geusen, "Development of modeling and simulation strategies for large-scale urban districts," in *Proc. Build. Simul.*, Sydney, Australia, 2011, pp. 1753–1760.

[21] S. Schutte and M. Sonnenschein, "Mosaik—Scalable smart grid scenario specification," in *Proc. Winter Simul. Conf. (WSC)*, 2012, pp. 1–12.

[22] S. Schutte, S. Scherfke, and M. Troschel, "Mosaik: A framework for modular simulation of active components in Smart Grids," in *Proc. IEEE 1st Int. Workshop Smart Grid Model. Simul. (SGMS)*, 2011, pp. 55–60.

[23] *Building Technologies Office: EnergyPlus Energy Simulation Software* (Feb. 2014) [Online]. Available: http://apps1.eere.energy.gov/buildings/energyplus/, accessed on Feb. 6, 2014.

[24] *Multi-Engineering Modeling and Simulation–Dymola–CATIA–Dassault Systèmes* [Online]. Available: http://www.3ds.com/products/catia/portfolio/dymola, accessed on Aug. 29, 2011.

[25] R. Mosshammer, F. Kupzog, M. Faschang, and M. Stifter, "Loose coupling architecture for co-simulation of heterogeneous components," in *Proc. 39th Annu. Conf. IEEE Ind. Electron. Soc. (IECON'13)*, 2013, pp. 7570–7575.

[26] M. Faschang, F. Kupzog, R. Mosshammer, and A. Einfalt, "Rapid control prototyping platform for networked smart grid systems," in *Proc. 39th Annu. Conf. IEEE Ind. Electron. Soc. (IECON'13)*, 2013, pp. 8172–8176.

[27] K. Anderson, J. Du, A. Narayan, and A. E. Gamal, "GridSpice: A distributed simulation platform for the smart grid," in *Proc. Workshop Model. Simul. Cyber-Phys. Energy Syst. (MSCPES)*, 2013, pp. 1–5.

[28] *MATLAB and Simulink for Technical Computing–MathWorks Deutschland* [Online]. Available: http://www.mathworks.de/, accessed on Feb. 14, 2013.

[29] *TLK-Thermo GmbH Engineering Services and Software for Thermal Systems* [Online]. Available: http://www.tlk-thermo.com/, accessed on Feb. 14, 2013.

[30] *NEPLAN® Power System Analysis and Engineering*, Zurich, Erlenbach, Switzerland [Online]. Available: http://www.neplan.ch/html/e/e_home.htm, accessed on Sep. 11, 2013.

[31] *IBM ILOG CPLEX Optimization Studio–Germany* [Online]. Available: http://www-03.ibm.com/software/products/de/de/ibmilogcpleoptistud/, accessed on Sep. 11, 2013.

[32] *ITI—Supporting Your Visions!: Home* [Online]. Available: http://www.iti.de/, accessed on Aug. 29, 2011.

[33] *Energiesimulationen–EnergieArchitektur* [Online]. Available: http://www.ea-energiearchitektur.de/leistungen/energiesimulationen, accessed on Sep. 11, 2013.

[34] R. Unger *et al.*, "'Green Building'—Modelling renewable building energy systems and electric mobility concepts using Modelica," presented at the 9th Modelica Conference, Munich, Germany, 2012.

[35] *Standard for Information Technology—Portable Operating System Interface (POSIX)*, IEEE Std. 10031-2001, 2001.

[36] *The Open Group Base Specifications Issue 7, 2013 Edition* [Online]. Available: http://pubs.opengroup.org/onlinepubs/9699919799/, accessed on Sep. 18, 2013.

[37] *Wetter und Klima–Deutscher Wetterdienst—Klimadaten* [Online]. Available: http://www.dwd.de/TRY, accessed on Sep. 12, 2013.

[38] G. Kerber, "Aufnahmefähigkeit von Niederspannungsverteilnetzen für die Einspeisung aus Photovoltaikkleinanlagen," dissertation, TU München, München, Germany, 2011.

Christoph Molitor (GSM'11) received the Dipl.-Ing. degree in electrical engineering from Rheinisch-Westfälische Technische Hochschule (RWTH) Aachen University, Aachen, Germany, in 2010, where he is currently pursuing the Ph.D. degree in electrical engineering.

In August 2010, he joined as a Research Associate with the Institute for Automation of Complex Power Systems, E.ON Energy Research Center, RWTH Aachen University. During his diploma program, from September 2007 to May 2008, he was a Visiting Student with the University College, Cork, Ireland. His research interests include home energy systems and energy management algorithms for smart cities.

Stephan Groß (S'14) received the Bachelor degree in mechanical engineering from Rheinisch-Westfälische Technische Hochschule (RWTH) Aachen University, Aachen, Germany, in 2014, where he is currently pursuing the Master degree in mechanical engineering with specialization in energy engineering.

Since 2011, he has been working as a Student Research Assistant with the Institute for Automation of Complex Power Systems, E.ON Energy Research Center, RWTH Aachen University.

Jakob Zeitz (S'14) received the Bachelor degree in business administration and industrial engineering from Rheinisch-Westfälische Technische Hochschule (RWTH) Aachen University, Aachen, Germany, in 2013, where he is currently pursuing the Master degree in business administration and engineering electrical energy technology.

Since 2011, he has been a Student Research Assistant with the Institute for Automation of Complex Power Systems, E.ON Energy Research Center, RWTH Aachen University.

Antonello Monti (M'94–SM'02) received the M.S. and Ph.D. degrees in electrical engineering from Politecnico di Milano, Milano, Italy, in 1989 and 1994, respectively.

From 1990 to 1994, he was with the Research Laboratory of Ansaldo Industria of Milan, Milano, designing the digital control of large power drives. In 1995, he joined as an Assistant Professor with the Department of Electrical Engineering, Politecnico di Milano. From 2000 to 2008, he was an Associate Professor and then a Full Professor with the Department of Electrical Engineering, University of South Carolina, Columbia, SC, USA. He is currently the Director of the Institute for Automation of Complex Power Systems, E.ON Energy Research Center, RWTH Aachen University, Aachen, Germany.