A Distributed Platform for Multi-modelling Co-simulations of Smart Building Energy Behaviour

Original
A Distributed Platform for Multi-modelling Co-simulations of Smart Building Energy Behaviour / Schiera, DANIELE SALVATORE; Barbierato, Luca; Lanzini, Andrea; Borchiellini, Romano; Pons, Enrico; Bompard, ETTORE FRANCESCO; Patti, Edoardo; Macii, Enrico; Bottaccioli, Lorenzo. - (2020). (Intervento presentato al convegno IEEE International Conference on Environment and Electrical Engineering (EEEIC 2020) tenutosi a Madrid, Spain nel 09-12 June 2019) [10.1109/EEEIC/ICPSEurope49358.2020.9160641].

Publisher:
IEEE

Published
DOI:10.1109/EEEIC/ICPSEurope49358.2020.9160641

A Distributed Platform for Multi-modelling Co-simulations of Smart Building Energy Behaviour

Daniele Salvatore Schiera*, Luca Barbierato*, Andrea Lanzini*, Romano Borchiellini*, Enrico Pons*,
Ettore Francesco Bompard*, Edoardo Patti*, Enrico Macii* and Lorenzo Bottaccioli*
*Energy Center Lab, Politecnico di Torino, Turin, Italy. Email: name.surname@polito.it

Abstract—**Nowadays, buildings are responsible of a large consumption of energy in our cities. Moreover, buildings can be seen as the smallest entity of urban energy systems. On these premises, in this paper we present a flexible and distributed co-simulation platform that exploits a multi-modelling approach to simulate and evaluate energy performance in smart buildings. The developed platform exploits the *Mosaik* co-simulation framework and implements the Functional Mock-up Interface (FMI) standard in order to couple and synchronise heterogeneous simulators and models. The platform integrates: *i*) the thermal performance of the building simulated with EnergyPlus, *ii*) the space heating and hot water system modelled as an heat pump with PID control strategy in Modelica, and *iii*) different Python models used to simulate household occupancy, electrical loads, roof-top photovoltaic production and smart meters. The platform guaranties a plug-and-play integration of models and simulators, hence, one or more models can be easily replaced without affecting the whole simulation engine. Finally, we present a demonstration example to test the functionalities and capabilities of the developed platform, and discuss future developments of our framework.**

Index Terms—**Co-simulation, Functional Mock-up Interface, Functional Mock-up Unit, Mosaik, Building Energy System, EnergyPlus, Modelica, Complex System.**

I. INTRODUCTION

Urban Energy System (UES) largely contribute to climate change due to high levels of energy consumption and greenhouse gas emission [1]. According to the United Nations Habitat, cities consume about 78% of global primary energy and generate more than 60% of greenhouse gas emissions primarily through consumption of fossil fuels for energy supply and transportation [2]. To engage these issues, designing and evaluating energy policies, improving energy efficiency, and integrating planning and operational dimensions of Renewable Energy Sources (RES) are keys to achieve a low carbon future. UES falls under the concept of Multi-Energy Systems (MES).

MES are complex systems where heterogeneous energy vectors (e.g. electricity, heat exchanging fluids, natural gas) interact together in such a multi-faceted way that they are very difficult to be analysed comprehensively [3]. MES complexity is difficult to be understood without exploiting models merging physical, economical, and social perspectives. Such analysis must keep into account not only constraints and feedback from regulators and economic drivers but also social and environmental behaviours. Therefore, the design, deployment, and management of UES need a holistic analysis with a multi-modelling approach to be effective. In this regard, co-simulation has been widely applied to integrate several models in order to represent and describe the complexity of urban energy systems [4]–[7]. Buildings play a key role as they are responsible of roughly 40% of the overall energy consumption [8] and can be considered as the smallest entity of a larger UES. Moreover, the shift to smart buildings offers great potential in energy-saving and grid balancing. For these reasons, they have attracted many researchers in developing co-simulation platforms to integrate models to describe the multi-faceted building complexity [9]–[15].

In this paper, we present a flexible and distributed co-simulation platform that exploits a multi-modelling approach in order to simulate and assess energy performance in smart buildings. The presented platform integrates heterogeneous simulation models by exploiting *Mosaik* Framework [16] that has been extended to embed also Functional Mock-up Interface (FMI) [17] allowing the interoperability among different simulation engines and tools. Thus, the proposed solution enables a modular infrastructure where different models can be integrated in a plug-and-play fashion. With respect to state of the art solutions, the proposed platform simulates the overall energy demand and behaviour of a smart building. The reviewed works focus only on thermal demand and household occupancy. Whilst, our solution integrates both models for household occupancy and thermal demand together with models to realistically replicate appliances' load consumption, photovoltaic production and smart meters. Finally, it integrates weather information provided by third-party services through web-service communication. Hence, co-simulation results are able to provide a holistic representation of energy usage in a smart building.

The structure of this paper is as follows. Section II explains the technological background of the framework proposed with a description of the main characteristics and their implementations. Section III provides a review of co-simulation of energy systems with a particular focus on building scale. Section IV shows the building energy co-simulation infrastructure with the description of the integrated models. Section V presents the case study used to demonstrate the capabilities of the proposed solution. Finally, Section VI reports concluding remarks and future works.

II. TECHNOLOGICAL BACKGROUND

The framework presented in this paper lays the foundations of a hybrid multi-modelling platform able to integrate subsystems of a UES coming from different specific domains of knowledge (i.e. component, communication, information, functions and business). The three main characteristics and strengths of the framework are outlined in the following.

Co-simulation Approach: Co-simulation has been identified as a flexible approach to simulate different dynamic models in a shared simulation environment. With this approach, it is possible to integrate multiple models describing a UES such as: *i*) users' activity and behaviour, *ii*) demand and supply, *iii*) control strategies, *iv*) economical strategies, *v*) RES, and *vi*) distribution networks [18]. Essentially, co-simulation is an approach for integrating system of systems, each one simulated by a different simulator engine (or solver). Therefore, domain-specific subsystems are addressed by their specialised solvers and their modelling tools. The approach preserves high efficiency and accuracy on simulations of the subsystems, which are coupled to obtain a more complex dynamic system of systems simulation.

FMI Standard: Commonly, domain-specific energy modelling tools and their solvers are not able to communicate and exchange information with each other. Moreover, they do not exchange information among different instances of the same simulation engine. To face this issue, Functional Mock-up Interface (FMI) is a tool-independent standard that allows *i*) to encapsulate models and their simulation engines, *ii*) to support direct control of the simulation, and *iii*) to exchange data among the models [17]. In practice, FMI defines an interface based on a set of C-functions composing the model and it is implemented by an executable, called Functional Mock-up Unit (FMU). The shared simulation environment interacts with a FMU through C-functions to create one or more instances of a model, to manage simulation evolution, and to exchange data enabling interaction among different simulation engines. FMU is a ZIP file that contains the model, its resources, documentation, and an XML file that describes the model structure and capabilities. Generally, FMU either has its solver, or it requires the simulation environment to perform numerical integrations.

The co-simulation infrastructure presented in this paper has been developed to permit the integration of models encapsulated in FMUs and their management through the FMI standard.

Co-simulation Orchestration: The co-simulation approach requires a master algorithm to create instances of models and orchestrate them in a shared and distributed simulation environment. In the past years, different co-simulation frameworks have been developed [18] based on specific application cases. The open-source co-simulation framework *Mosaik* [16] provides good performance, high usability, and flexibility. Indeed, *Mosaik* can be integrated with several power grid simulators (e.g. python simulator, PYPOWER, Opal-RT, PowerFactory) and any other simulators thanks to FMU integration (e.g. MatSim, Modelica, EnergyPlus, MATLAB) through the adaptation of *Mosaik APIs*. *Mosaik* manages the simulation time-step of all models, software and simulators, and permits the exchange of data between them. The time synchronisation is based on a *Discrete Event* synchronisation method. To do so, *Mosaik* sets a schedule based on the time-step description provided by each simulator. Accordingly to these descriptions, the schedule contains pre-defined synchronisation points and exploits a directed acyclic schedule graph to determine the order of step commands, which are sent to every simulator. For all the reasons explained above, *Mosaik* was chosen as the Co-simulation Orchestration Engine (COE) in our distributed co-simulation platform.

III. RELATED WORKS

In the last few decades, a robust research effort has been given to develop domain-specific simulation tools that have been used to simulate the behaviour of energy systems and solve the problems of a particular domain with high efficiency and accuracy [19]. A growing effort appears to focus on combining two or more modelling frameworks to integrate aspects of different specific domains and functional layers with the exploitation of novel methodologies, standards, and tools [18], such as co-simulation platforms. Co-simulation platforms have been developed for studying applications such as new strategies for city energy supply and demand, urban energy planning, or distribution networks analysis and stability for an efficient RES penetration in a new smart citizen-centric energy system [6], [7]. In [6], authors demonstrated that Functional Mock-up Interface is essential in co-simulating multi-physical district models. In particular, they performed a comparative analysis between canonical integrated simulation and co-simulation through FMI with the aim of assessing performance and scalability. They demonstrated that co-simulation can run up to 90 times faster than the integrated simulation for a 24-dwellings district. Meanwhile, authors in [7] showed the capability of the *Mosaik* framework that have been used to develop a Cyber-Physical Energy Systems (CPES) test environment for simulation planning, uncertainty quantification and the development of multi-agent systems. The great potential in energy-saving and grid balancing that smart buildings could offer has attracted many researchers in developing co-simulation frameworks tools able to describe the various building elements and their interaction. In [9] a framework to couple a multi-agent stochastic simulation of occupants with a building performance simulation tool based upon EnergyPlus [20] was presented. The aim was to merge the stochastic nature of occupants presence, activities and behaviours into building simulation software in a coherent and generalised way. Thomas et al. [10] focused on a multi-scale coupling process through FMI to enable a co-simulation between *i)* an energy model of a building simulated with EnergyPlus and *ii)* an energy model of a city simulated with CitySim. The authors compared the coupled and uncoupled simulations highlighting strengths and weakness of the tool-chain. In particular, they obtained a close correlation between the coupled and uncoupled EnergyPlus simulation, while they noted a wider discrepancy between the CitySim results. Moreover, in [11] a framework capable of performing a model predictive control (MPC) through real-time Building Energy Management System (BEMS) data was presented. The aim of the co-simulation platform is to test the effectiveness of the MPC in reducing energy consumption and achieve the desired temperature comfort. As a result, they obtained a high prediction rate, showing energy-saving benefits applying a MPC strategy to BEMS. In [12], authors proposed a co-simulation approach to couple systems based on waveform relaxation method. The work tries to reduce the computation time by reducing the number of calls of the different sub-models. The proposed approach was performed on an energy building system over FMU components and web services, making a comparison with the classical coupling methods. They demonstrated that the approach has good performances when the simulation time is longer, but the method does not have a high efficiency on hysteresis models. In [13], an occupant behaviour modelling tool encapsulated into an FMU allows a co-simulation with building energy modelling programs. The tool implements also interoperability among occupant behaviour models. In [14] the FMI standard was exploited to couple a building energy performance simulator with a Modelica-based HVAC system and plant models. The authors tested different co-simulation algorithms in order to provide suggestions on choosing a suitable one based on accuracy and simulation performance. Finally, authors of [15] developed a proof of concept urban energy co-simulation framework based on FMI and *Mosaik*. The framework was applied coupling two simulation tools, EnergyPlus and NO-MASS, showing that the results were as expected and accurate in comparison with the framework results obtained by [9].

To the best of our knowledge none of the reviewed platforms, tools and framework co-simulate the overall elements of a building. They mainly focus in coupling occupancy model with thermal models, or thermal models with external heating/cooling system models, or thermal models with MPC to test the effectiveness of the strategy. Hence, none of these literature solutions couple together users' occupancy, the thermal and electrical energy demand and production of a building. With respect to such literature solutions, our platform integrates different heterogeneous and distributed models to realistically simulate: *i)* users' occupancy, *ii)* electrical appliances and their energy consumption, *iii)* heating/cooling system with integrated control strategies, *iv)* PV energy production based on geographic information system, *v)* thermal behaviours in buildings, *vi)* smart meters. The platform also integrates third-party and services, available on the web, to retrieve real meteorological data and forecasts. Moreover, the use of *Mosaik* as COE offers high flexibility and performance to our platform, allowing integration of numerous domain-specific models and simulators directly connected to COE and/or through the FMI/*Mosaik* adapter. Indeed, we closely followed the frameworks of [7], [15] that implemented an adapter to support the FMI standard into *Mosaik*. It is worth noting that combining both the *Mosaik* framework and the FMI standard enables a complete framework where the overall multi-models co-simulation engine is spread across different servers and computers. This increases the scalability of our platform, even in easily integrating and/or replacing other simulation models, tools or simulators in a plug-and-play fashion.

IV. CO-SIMULATION PLATFORM

The presented platform uses a co-simulation approach to integrate different domain-specific subsystems, tools and simulators. To this purpose, we extended the *Mosaik* framework [16], which is a flexible COE, embedding the FMI co-simulation standard [17]. Such extension is required to enable the integration of simulation models with their solver engines that are not directly controllable by the *Mosaik APIs*. Fig. 1 shows the scheme of the proposed framework, highlighting the main layers of the platform, which are described in the following:

i) Scenarios and Data I/O. *Scenarios* contain information about configuration set up of the simulators to be included, i.e., topology to connect model instances of these simulators, their time-steps, initial conditions, and data flow between model instances. In addition, *Data I/O* module provides an interface to configure the shared simulation environment and collect the data requested.

ii) Mosaik COE. The COE is mastered by *Mosaik* framework in Python programming language, providing the ability to orchestrate the simulation and manage the data flow of each model instance. As shown in Fig. 1, *Mosaik Scenario APIs* and *Mosaik Simulator APIs* are used to set the COE. *Mosaik Scenario APIs* starts simulators and instantiate models from them exploiting information retrieved from the selected *Scenario*. Meanwhile, *Mosaik Simulator APIs* defines essential interfaces, enables model instances to exchange information and executes all stages of the simulation process through the orchestrator (i.e., instantiation, initialization, do step, set and get data). Finally, the COE manages the simulation time-step, synchronising the time and data flow between all models.

iii) FMPy and FMI/Mosaik adapter. FMPy is a Python library for loading and interacting with FMUs using native Python language, thus providing a Python interface to the FMI standard making interaction with FMUs through FMI functions. Hence, an FMI/*Mosaik* adapter was developed in

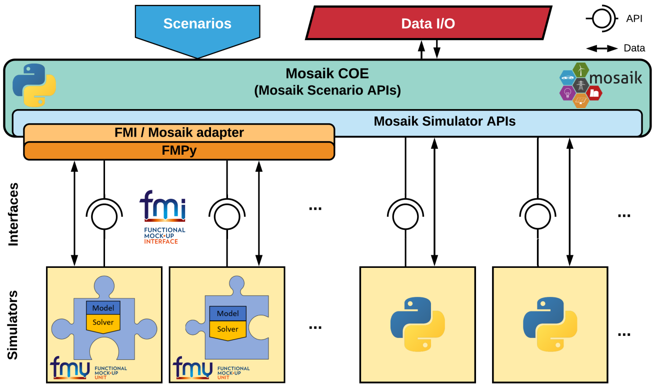

Fig. 1. Scheme of our multi-modelling co-simulation framework.

order to map *Mosaik Simulator APIs* and FMI functions through FMPy library.

iv) Simulators and Interfaces. Simulators are represented in Fig. 1 as blocks that are connected to the COE through data exchange interfaces and APIs. Thanks to the flexibility given by *Mosaik* and the FMI/*Mosaik* adapter, the co-simulation platform can accept encapsulated models as FMUs and/or Python simulators directly.

A. *Scenario making on co-simulation platform*

In this section, we describe a scenario of an energy building system made by the connections of different simulators in order to demonstrate the application of the proposed co-simulation platform and, subsequently in Section V, we show how the described system is used in a demonstration example.

Fig. 2 illustrates the scheme of model blocks, focusing on the energy and data flow between blocks. In this scenario, an EnergyPlus building model is implemented in the platform as an FMU, and linked to a Modelica-based Electric Heat Pump system, encapsulated in an FMU as well. Furthermore, a photovoltaic system, the household electricity behaviour, and weather model are provided to the building by linking external Python simulators. Each block in Fig. 2 is explained as follows:

i) Solar & Local Weather. Weather data are integrated from third-party data sources, such as Weather Underground [21], through an integrated Python application interface. The application retrieves the information at the time step required by the models that need the data and distributes them in run-time through *Mosaik* that manages the time synchronisation. The weather data provided to the other models during simulation are the solar radiation (*GHI*), the outdoor dew temperature (*TDew*), the outdoor relative humidity (*RH*) and the outdoor dry bulb temperature (*TDryBul*).

ii) Photovoltaic system. The Photovoltaic (PV) system was modelled by integrating the simulation infrastructure presented in [22]. The infrastructure allows to estimate the PV potential and to simulate the solar radiation profiles in real-sky conditions with a high spatio-temporal resolution. It uses as inputs: *(a)* GIS data to describe building rooftops in terms of slope, orientation, possible obstacles, and shadows; *(b)* weather data provided by *Solar & Local Weather* block to calculate the incident solar radiation in real-sky condition (*GHI*). The on-site generated electricity is primarily self-consumed by the household while any surplus is sent to the grid.

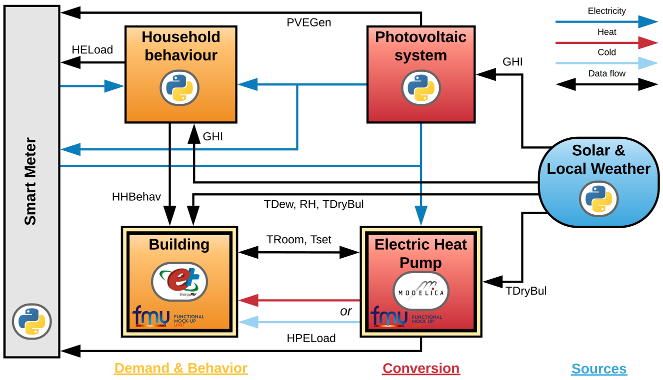

Fig. 2. Block diagram of the scenario built with the co-simulation platform. The diagram shows both the energy- and data-flow connections between simulation blocks.

iii) Household behaviour. The household electricity behaviour was integrated in the co-simulation platform by using the Python simulator proposed in [23]. The model uses different kind of input data to create a non-homogeneous semi-Markov model for simulating the household electricity behaviour and retrieve the aggregated electricity load profiles.

The model creates the household by specifying the composition of the family, starting from census data. Then, the set of appliances in the households are distributed according to statistics obtained from Use of Energy surveys. Whilst, the statistics obtained from the Time of Use surveys were used to create Semi-Markov model and generate each person behaviour. Finally, the simulator uses the created Semi-Markov model to generate household behaviour in terms of presences, type and duration of each activity performed by household's inhabitants, which is associated to specific usage of electric appliances (e.g., washing machine, dish-washer, vacuum cleaner, fridge, etc.). The simulator uses also weather data, provided by *Solar & Local Weather* block, to compute energy consumption of domestic lighting systems according to solar radiation (*GHI*).

At the end, the Household behaviour parses the number of occupants in a zone and interactions with lights and appliances, providing single person, aggregated electrical loads, appliances and lights loads and schedules (*HHBehav*), which are given as input to the *Building* block.

iv) Electric Heat Pump. The air-to-water Electrical Heat Pump (EHP) has been developed using standard components models in Modelica, and it was exported as an FMU, which contains the simulation model and exposes inputs and outputs, as shown in Fig. 2. The EHP was interfaced using the FMI functions through *Mosaik APIs*.

The EHP model computes the sensible heat gain required to maintain the set-point temperature (*Tset*) in rooms. The FMU needs as input the room temperature (*TRoom*), provided by the building model. The coefficient of performance of EHP is set by a parametric relationship with the outdoor temperature (*TDryBul*) and the outlet flow temperature of the water-based underfloor heating system. The parameters were taken by datasheets of common residential Heat Pumps. The most implemented regulation system for the outlet flow temperature is the use of a climatic curve, which sets the temperature based on the outdoor temperature through a piecewises linear function. The output of the EHP FMU is the heat requested by the building. The input variable *TRoom* is controlled to maintain the desired set-point *Tset* by implementing a Proportional-Integral-Derivative (PID) controller that acts on the water mass flow rate of the heating system.

v) Building. The building was modelled in EnergyPlus, a well-know detailed energy simulation tool that performs calculation of the building heating and cooling loads, disaggregated energy end-uses, Energy Management Systems and many other variables. Moreover, the *EnergyPlusToFMU* package written in Python can export the building simulation program as an FMU. Inputs and outputs of Building FMU are managed by EnergyPlus through three types of *External Interface* objects: *i)* the Household behaviour variables (*HHBehav*) and heat gain from the EHP were interfaced as input schedules; *ii)* the outdoor dry-bulb temperature (*TDryBul*), the outdoor air relative humidity (*RH*), and the outdoor dew temperature point (*TDew*) were interfaced as input actuators; *iii)* the indoor temperature (*TRoom*) and set point temperature (*Tset*) were linked to external interface as output variables.

vi) Smart Meter. The virtual Smart Meter provides the physical and data interface between the building system and the distribution network. It receives the PV electrical generation (*PVEGen*), the aggregated household electrical load (*HELoad*) and the EHP electrical consumption (*HPELoad*). It was used as a data collector manager that returns and shows the simulation results either in run-time or at the end of the simulation.

V. SCENARIO SIMULATION

To test functionalities and capabilities of the presented co-simulation platform, the scenario proposed in Section IV-A is applied to a hypothetical and realistic house located in Turin, Italy.

Our test-case consists of a *Building* of about 150 m^2 (see Fig. 3) modelled and simulated in EnergyPlus. Materials and constructions layers of the Building model have been defined following the most frequent composition and layout schemes of the existing Italian building stock. The *Solar & Local Weather* block provides weather data to the sub-models of the scenario giving the location. Before starting the co-simulation, the *Household behaviour* block generates the family, starting from the local socio-demographics and energy-related data. The generated family is composed by four people: a full-time man worker, an housewife, and two student kids.

The PV system on the rooftop is south oriented with a 37° tilt angle. It provides 5 kWp to the house. To satisfy the heating demand of the house, an Electrical Heat Pump is installed with a power input of 5.5 kWe and a COP of 4 at nominal conditions. During the winter season, the desired indoor temperature was set to 20° C, according to the Italian directive. After creating the models, the simulation blocks are instantiated using the *Mosaik Simulator APIs* (defining parameters, constants, and time-step), and connected to each other via *Mosaik Scenario APIs* (as shown in Fig. 2). The time-steps Δt were set considering the scenario characteristics, capabilities of the solvers, and computational effort: EnergyPlus (i.e. the Building block in Fig. 2) 10 min, Modelica (i.e. the Electric Heat Pump) 5 min, PV simulator 15 min and Household behaviour 10 min. The weather simulator provides data to each simulation engine at the requested time-step.

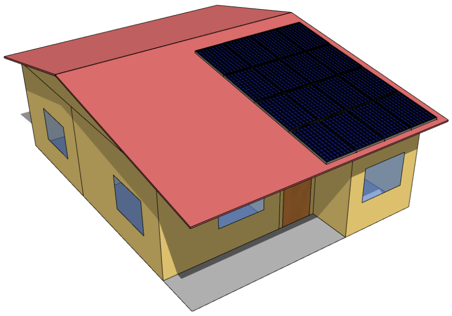

Fig. 3. The layout of the house modelled in EnergyPlus.

The results of the simulation are shown in Fig. 4 and Fig. 5, showing a general time window of 7 days in the winter season. From our simulation framework, we can extract information from either each module that exposes inputs/outputs and parameters, or simulation environment directly. The data were collected every 10 $minutes$.

During a winter day, the outdoor temperature can quickly swing, as shown in Fig. 4-(b). The requested heating demand is satisfied by the EHP and, thanks to the PID control strategy, the room temperature remains around 20° C with only little oscillations with an amplitude of about 0.75° C, as shown in Fig. 4-(a). The PID controller was tuned during run-time simulation till reaching optimum values and stability for the desired control response. Indeed, the tunable parameters can be changed during execution, thus directly seeing the feedback results on simulation.

The power-related time-series are depicted in Fig. 5: Fig. 5-(a) shows the net power required from the grid (difference between the total loads and PV production) in relation to the occupancy level of inhabitants, Fig. 5-(b) reports the desegregated loads with respect to appliances, lights, and EHP consumption, and Fig. 5-(c) shows the PV production.

As we can see, the use of appliances and lights follows the presence of inhabitants and the daily cycle as well. Furthermore, comparing the curves of EHP load and outdoor temperature, we can observe that the EHP consumption increases when the outdoor temperature drops and vice versa. This is a typical behaviour linked to the outlet flow temperature of the EHP that is regulated by a climatic curve. Moreover,

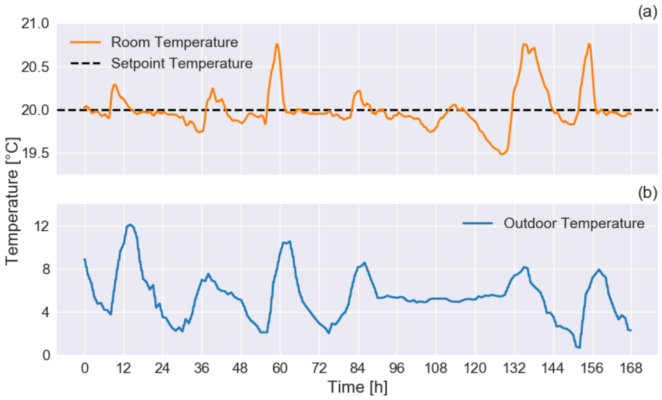

Fig. 4. Indoor (*a*) and Outdoor (*b*) temperature trends during winter days.

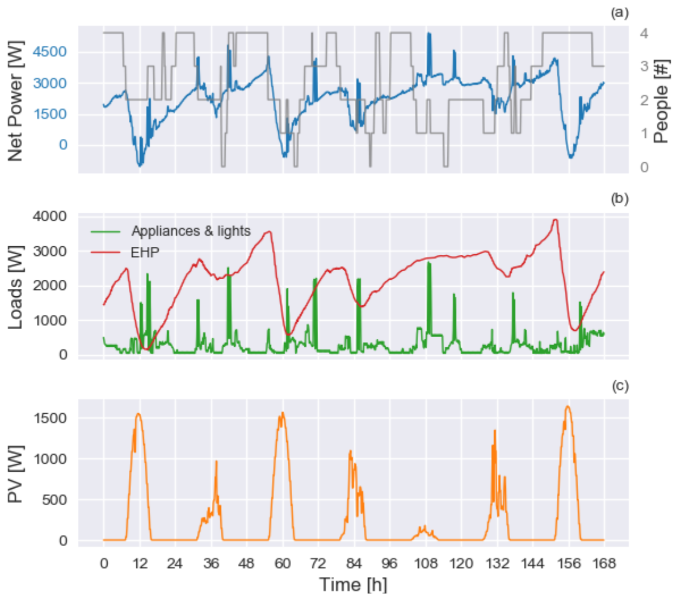

Fig. 5. (*a*) Net Power and occupancy level, (*b*) desegregated loads and (*c*) PV production.

the electric load requested by the EHP to satisfy the heating demand is considerably higher than the electrical household consumption due to appliances and lights. Therefore, the net power almost follows the EHP load.

Generally, during winter days, the on-site electricity production does not reach significant levels able to cover the EHP load always. Moreover, as shown in Fig. 5, the daily maximum peaks of the PV power matches with the daily minimum peaks load of the EHP, thus reducing the self-consumption. As a consequence, the net power can be negative and the excess of generated electricity is sent into the grid.

VI. CONCLUSION AND FUTURE WORKS

In this work, we presented a distributed co-simulation platform to perform energy assessments in Smart Buildings. The platform permits to integrate several and heterogeneous simulators by coupling *Mosaik* framework and the FMI standard. From the results, it is possible to see how with the proposed approach we can co-simulate different aspects of a building with an high-level of detail.

In future works, we intend to extend our framework to co-simulate an entire urban district with its distribution networks. Moreover, we are planning to replace software models of physical components with real devices (e.g. smart meters and PV systems) unlocking real-time Hardware-In-the-Loop (HIL) co-simulations. An open challenge in unlocking such HIL functionalities consists on addressing hard synchronisation requirements among models and simulators or, more in general, among hardware and software components in the co-simulation environment to guarantee the correctness of simulations.

REFERENCES

[1] U. DESA, “World urbanization prospects: The 2018 revision,” 2019.

[2] United Nations, “Energy, UN-Habitat,” https://unhabitat.org/urban-themes/energy/.

[3] P. Mancarella, “MES (multi-energy systems): An overview of concepts and evaluation models,” *Energy*, vol. 65, p. 1–17, 02 2014.

[4] L. Bottaccioli, A. Estebsari, E. Pons, E. Bompard, E. Macii, E. Patti, and A. Acquaviva, “A flexible distributed infrastructure for real-time co-simulations in smart grids,” *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 3265–3274, 2017.

[5] D. S. Schiera, F. D. Minuto, L. Bottaccioli, R. Borchiellini, and A. Lanzini, “Analysis of rooftop photovoltaics diffusion in energy community buildings by a novel gis-and agent-based modeling co-simulation platform,” *IEEE Access*, vol. 7, pp. 93 404–93 432, 2019.

[6] V. Reinbold, C. Protopapadaki, J.-P. Tavella, and D. Saelens, “Assessing scalability of a low-voltage distribution grid co-simulation through functional mock-up interface,” *Journal of Building Performance Simulation*, pp. 1–13, 2019.

[7] C. Steinbrink, M. Blank-Babazadeh, A. El-Ama, S. Holly, B. Lüers, M. Nebel-Wenner, R. P. Ramírez Acosta, T. Raub, J. S. Schwarz, S. Stark *et al.*, “Cpes testing with mosaik: Co-simulation planning, execution and analysis,” *Applied Sciences*, vol. 9, no. 5, p. 923, 2019.

[8] European Parliament, “Directive 2010/31/EU of the European Parliament and of the Council of 19 May 2010 on the energy performance of buildings,” 2010.

[9] J. Chapman, P.-o. Siebers, and D. Robinson, “Coupling multi-agent stochastic simulation of occupants with building simulation,” in *In Proc. of: BSO14*, 2014.

[10] D. Thomas, C. Miller, J. Kämpf, and A. Schlueter, “Multiscale co-simulation of energyplus and citysim models derived from a building information model,” in *In Proc. Of: IBPSA Bausim 2014*, 2014, pp. 469–476.

[11] Y. Kwak, J.-H. Huh, and C. Jang, “Development of a model predictive control framework through real-time building energy management system data,” *Applied Energy*, vol. 155, pp. 1–13, 2015.

[12] A. Raad, V. Reinbold, B. Delinchant, and F. Wurtz, “Energy building co-simulation based on the wrm algorithm for efficient simulation over fmu components of web service,” *Proceedings of the BS*, vol. 15, 2015.

[13] T. Hong, H. Sun, Y. Chen, S. C. Taylor-Lange, and D. Yan, “An occupant behavior modeling tool for co-simulation,” *Energy and Buildings*, vol. 117, pp. 272–281, 2016.

[14] A. Nicolai and A. Paepcke, “Co-simulation between detailed building energy performance simulation and modelica hvac component models,” in *In Proc. of: 12th International Modelica Conference, Prague, Czech Republic, May 15-17, 2017*, no. 132. Linköping University Electronic Press, 2017, pp. 63–72.

[15] K. Wang, P.-O. Siebers, and D. Robinson, “Towards generalized co-simulation of urban energy systems,” *Procedia engineering*, vol. 198, pp. 366–374, 2017.

[16] OFFIS - Institute for Information Technology, “Mosaik,” https://mosaik.offis.de/.

[17] Modelica Association Project, “Functional Mockup Interface,” https://fmi-standard.org/.

[18] M. Vogt, F. Marten, and M. Braun, “A survey and statistical analysis of smart grid co-simulations,” *Applied energy*, vol. 222, pp. 67–78, 2018.

[19] H.-K. Ringkjøb, P. M. Haugan, and I. M. Solbrekke, “A review of modelling tools for energy and electricity systems with large shares of variable renewables,” *Renewable and Sustainable Energy Reviews*, vol. 96, pp. 440–459, 2018.

[20] U.S. Department of Energy’s (DOE) Building Technologies Office (BTO), “Energy Plus,” https://energyplus.net/.

[21] Weather Underground, “Weather Forecast & Reports, Long Range & Local,” https://www.wunderground.com.

[22] L. Bottaccioli, E. Patti, E. Macii, and A. Acquaviva, “Gis-based software infrastructure to model pv generation in fine-grained spatio-temporal domain,” *IEEE Systems Journal*, vol. 12, no. 3, pp. 2832–2841, 2017.

[23] L. Bottaccioli, S. Di Cataldo, A. Acquaviva, and E. Patti, “Realistic multi-scale modeling of household electricity behaviors,” *IEEE Access*, vol. 7, pp. 2467–2489, 2018.