



# Distributed Co-Simulation Applied To Urban Scale Energy Systems Design

Pablo Puerto, Jessen Page, Bruno Ladevie, Jakob Rager

## ► To cite this version:

Pablo Puerto, Jessen Page, Bruno Ladevie, Jakob Rager. Distributed Co-Simulation Applied To Urban Scale Energy Systems Design. IBPSA 2019 - 16th Conference of the International-Building-Performance-Simulation-Association, Sep 2019, Rome, Italy. pp.1365-1371, 10.26868/25222708.2019.210491 . hal-03474697

HAL Id: hal-03474697

<https://imt-mines-albi.hal.science/hal-03474697>

Submitted on 10 Dec 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

## Distributed co-simulation applied to urban scale energy systems design

Pablo Puerto<sup>1,2,3</sup>, Jessen Page<sup>2</sup>, Bruno Ladevie<sup>3</sup>, Jakob Rager<sup>1</sup>

<sup>1</sup>CREM, Martigny, SWITZERLAND

<sup>2</sup>HES-SO, Sion, SWITZERLAND

<sup>3</sup>IMT Mines Albi / UMR CNRS 5302, Albi, FRANCE

### Abstract

Urban energy systems are now considered as multi-network structures rather than as silo-like separated energy vectors. However, multi-network structures imply a growing number of intricate interactions between various components.

The design and the operation of such systems require taking into account complex multi-domain operational constraints. The aim of this work is to allow for a detailed and distributed simulation of the dynamic behaviour of coupled energy systems at urban scale for design and decision support purposes.

With a development based on open source cutting-edge information technologies tools and applications, this paper proposes an innovative framework providing simplified deployment, distributed computation and encapsulation for enhanced and reproducible co-simulation applied to the simulation based validation of a design for a complex urban energy system.

### Introduction

Considering urban energy systems as complex multi-network structures rather than the more classical silo-like approach of separated energy vectors has become widely accepted in the scientific literature. This notion has been examined from different and complementary perspectives, such as shown from the electrical (Ilic et al. (2010)) and from the thermal (Lund et al. (2014)) point of view or for hybrid networks (Widl et al. (2018)).

Urban energy systems conceived as multi-network structures imply a growing number of intricate interactions between various components. The design and the operation of such systems requires taking into account complex multi-domain operational constraints. Simulation is a compulsory step to define and to practically implement complex systems. Finding a viable compromise between integrating the wanted level of detail in the modelling process – in order to consider critical operational constraints – and ensuring an acceptable computation time is one of the biggest challenges in the simulation of urban energy systems (van Beuzekom et al. (2015)). Furthermore, it re-

quires competence over multiple domains like natural gas distribution, medium- and low-voltage power grid regulation or district heating systems operation.

Using different tools and models for each of the components of such a complex system allows to benefit from the specific features offered by those tools in each domain and to capitalise on the experience of specialised users. Within this context, co-simulation can be defined as the coupling of simulation tools (also referred to as simulators) for assessing a partitioned complex system.

The idea is to partition a system into sub-systems, modelling each of them separately with a specific simulator and to re-create the global behaviour of the complete system by exchanging data between the models of the sub-systems at simulation runtime. A co-simulation tool is the software synchronising execution and data exchange of the individual simulators coupled within a co-simulation. The connections and links between the simulators modelling the sub-systems form a co-simulation graph.

A co-simulation graph can be represented as a directed graph with the simulators (and their models) of the sub-systems as nodes and links connecting model inputs and outputs as edges. Each node represents an individual technical component or a group of technical components (a building with its decentralised heat production, a centralised heat pump, a power grid, etc.).

The simulation based validation of an urban scale energy system design allows to: (1) define the impact on existing infrastructures, (2) understand the dynamic behaviour of coupled systems, (3) assess and compare control strategies with the simulation used as a virtual test bed.

This work is part of the IntegrCiTy<sup>1</sup> project that focuses on the development and implementation of an integrated decision-support environment for city planners and energy providers to improve the efficiency and the resilience of energy supply infrastructures. An important asset in this decision support environment is a co-simulation framework, which al-

<sup>1</sup><http://iese.heig-vd.ch/projets/integricity>

lows to perform very detailed technical assessments of proposed changes and extensions of energy supply infrastructure.

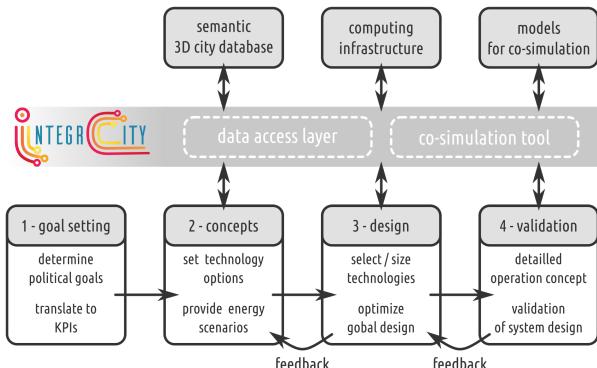


Figure 1: The IntegrCiTy structure.

This paper is organised as follows: the developed co-simulation framework along with its features and implementation will be first discussed. Then the system that will be used as an example of application for the co-simulation framework will be described, along with its different sub-systems design and modelling processes. The results of the use case example will be presented after and followed by a discussion on the application of such an approach to the design of urban energy system. Finally the conclusions are reported in the last section.

## Co-simulation framework

### Related work

This section provides an overview of previous literature and related work on the topic of frameworks and tools for heterogeneous modelling approaches especially suited for the co-simulation of urban scale energy systems.

The research and development of tools for this topic is quite prolific. Multiple examples of frameworks already exists such as Ptolemy II (Eker et al. (2003)), BCVTB (Wetter and Haves (2008)) which uses Ptolemy II, Mosaik (Schutte et al. (2011)) based on Python, OpenBuildNet (Nghiem et al. (2016)) developed in C++ with a version allowing to use encapsulation, and FUMOLA (Widl et al. (2015)) using Ptolemy II and FMIPP<sup>2</sup> for FMI<sup>3</sup> integration.

The orchestration processes are already well developed and mature. They propose various communication patterns from serial (Gauss-Seidel) to parallel (Jacobi) or even hybrid solutions.

However the definition and the deployment of the defined co-simulation may be enhanced as well as the encapsulation of the different processes running the simulation of the sub-systems models. This encapsulation allows to improve the re-usability by creat-

ing dedicated, independent and reproducible computation environments.

The naturally distributed structure of a co-simulation can be used to distribute computation over multiple computers. The co-simulation approach combines simulation tools that may run on different operating systems, or may use different versions of dependencies. Those two aspects create the need for a framework able to ensure the encapsulation, the distribution and the communication between the distributed simulation processes.

The main scientific innovation of this study resides in the co-simulation framework called ZerOBNL developed using open source cutting-edge information technologies tools and applications compared to existing frameworks described in Vogt et al. (2018), especially the combined use of Docker containers technology with ZeroMQ communication capabilities.

The co-simulation framework provides (1) semantics and associated functionality for the partitioning and the modelling of the system under study as well as for the setup of the co-simulation, (2) a master process based on a communication schema for the orchestration of the co-simulation execution and (3) a solution to create and deploy independent, dedicated, reusable, reproducible and dispatchable environments for simulation tools and models.

The dedicated conceptual and semantic layers as well as the deployment and results retrieval features have been integrated into a compact application programming interface delivered as a Python library.

### Abstraction and semantic layer

A sub-system is one block of the partitioned studied system that will be simulated using a dedicated model implemented in a simulation tool.

The concept of *node* is the central concept of the framework. It is associated with a instance of a sub-system model. A *node* is defined with the associated files needed for the simulation, parameters and initial values for the model attributes. A *meta-model* and an *environment* are also assigned to a *node*.

The concept of *environment* allows to determine (1) the dependencies and associated software needed by the simulation tool in charge of the simulation of the sub-system model via a Dockerfile described below and (2) the wrapper that implements the way to interface the simulation tool and the model with the communication mechanism and the master orchestration process, both described in the coming sections. The concept of *meta-model* defines the inputs and outputs of the model of the sub-system.

Finally a *link* allows to setup a data exchange during simulation. A *link* is defined by two [*node*, *attribute*] couples: one to get a value and one to set the value. *Attributes* also have a unit entry allowing when creating a *link* to check if the both ends of the *link* share

<sup>2</sup><https://sourceforge.net/projects/fmipp/>

<sup>3</sup><https://fmi-standard.org/>

the same physical unit.

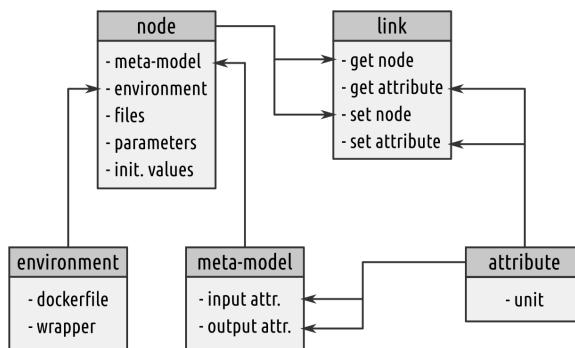


Figure 2: The proposed conceptual models.

### Orchestration master process

A hybrid approach has been chosen in order to avoid the two step overshoot effect from a pure Jacobi pattern but also to take advantage of the parallel computation opportunity made available by the software structure described below and by definition by the Jacobi method. This approach consists of a series of groups of parallel simulated nodes, the groups of nodes are run serially but within the group all the nodes are run in the same time. Once all the nodes of a group have achieved the current simulation step, every concerned link is activated and inputs attributes of other nodes are updated from the freshly computed outputs. Such a simulation pattern is not valid if the nodes of a group share a direct link otherwise the input value of a node can be updated after the simulation step of the node. This may cause a two step overshoot effect.

**Sequence = [ [Node A, Node B] , [Node C, Node D] ]**

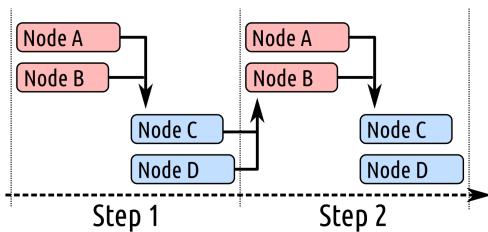


Figure 3: The proposed simulation sequence.

### Implementation and deployment

The co-simulation framework concept is mainly based on the combination of two technologies described below: Docker<sup>4</sup> for the virtualisation and the encapsulation of the multiple simulation processes and ZeroMQ<sup>5</sup> for the concurrency (multi-thread) and the communication between the simulation processes. The two technologies are available for all commonly used operating systems, based on Windows or Unix.

<sup>4</sup><https://www.docker.com/>  
<sup>5</sup><http://zeromq.org/>

ZeroMQ is a broker-less (fully decentralised) messaging system and with application programming interface available for C, C++, Go, Java, Lua and the chosen one for the co-simulation framework Python. This creates a platform and language-neutral framework allowing to connect tools from a large variety of programming languages over different operating systems.

By comparing multiple networking libraries Lilis et al. (2018) demonstrate the capacity of ZeroMQ to provide high performance communication and low induced overhead in terms of computation time.

The co-simulation framework relies on Docker to create dedicated and isolated containers that package up code and dependencies for each model of sub-systems. Docker allows to run containers quickly and reliably from one computing environment to another and provides a simple way to allow communication between containers by creating networks.

This encapsulation also drastically reduces memory usage (RAM) compared to other solutions (e.g., virtualization via VirtualBox) and makes it possible to run a co-simulation with a much larger number of containers as shown by Zhang et al. (2018).

The deployment of the containers hosting the simulation tools as well as the master orchestration process is done using a native Docker solution called Docker Compose<sup>6</sup>. Using a YAML<sup>7</sup> file, all the required containers and the inter-dependencies between them can be specified. This YAML file is created by the co-simulation framework on-the-fly before the simulation, based on the defined co-simulation scenario. It is then run with Docker Compose as a child process using Python's subprocess module<sup>8</sup>. Docker Compose gathers the logs of the creation and the execution of all the containers and stores them properly. Thus, once the full simulation process has finished, it is simple to spot errors and identify associated nodes.

Using Docker containers also allows to use a natively available tool called Docker Swarm<sup>9</sup> without any changes in the framework for distributed computation. It provides a way to automatically distribute running containers over multiple hosts, from two combined computers to more complex clusters maintaining communication channels' orchestration features out-of-the-box. The computation time saving potential induced by distributed computation for this particular application is very dependant on the partition of the modelled system, and more precisely on the number of nodes in the same group that can be run simultaneously. Docker Swarm also allows to combine hosts running different operating systems in the same co-simulation, which is useful for example for

<sup>6</sup><https://docs.docker.com/compose/>

<sup>7</sup><https://yaml.org/>

<sup>8</sup><https://docs.Python.org/3/library/subprocess.html>

<sup>9</sup><https://docs.docker.com/engine/swarm/>

running and communicating between Windows-only and Linux-only simulation tools.

The communication between the master orchestration process and the nodes goes through two different channels. The master orchestration process can publish messages and broadcast them to a group or to all the nodes via a publish/subscribe pattern. The nodes can then respond to the message by sending a response message to a first-in-first-out queue via a push/pull pattern.

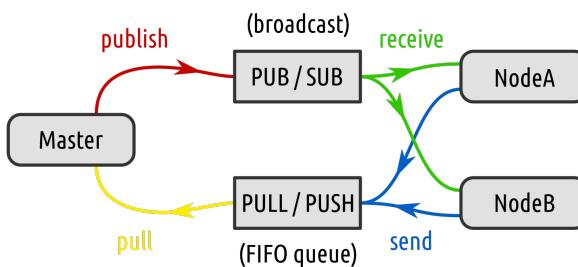


Figure 4: The proposed communication pattern.

During the simulation, all the values exchanged between the nodes are stored into a results database. This database is implemented into a Redis<sup>10</sup> in-memory data structure store running in a container. Even if the simulation crashes, all the results until the crash are safely stored and available.

The wrappers allowing the interaction between the master orchestration process and the simulation tools are developed in Python. The use of the FMIPP package allows to interact directly with FMI compliant tools<sup>11</sup>. The versatility of the Python language combined with the large ecosystem of existing libraries makes it a viable choice for interfacing with tools which do not have the ability to interface with FMI.

## Case study and modelling

### Studied system and partition

The developed co-simulation framework has been applied on a test case based on a proposed concept for the future energy system of a neighbourhood of Vevey in Switzerland.

This concept combines medium and low voltage power grids, medium and low pressure natural gas networks, a low-temperature district heating network (LT-DHN) based on centralised and decentralised heat pumps using a lake as the main thermal source and a gas-fired combined heat and power plant supplying two small scale high temperature district heating networks (HT-DHN). Two heat pumps, considered as consumers on the LT-DHN also supply heat to the two HT-DHN.

The power grid and the district heating networks are

<sup>10</sup><https://redis.io/>

<sup>11</sup><https://fmi-standard.org/tools/>

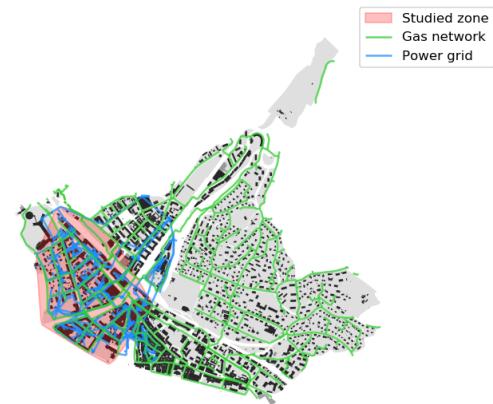


Figure 5: The studied area - Vevey.

coupled with the centralised and decentralised heat pumps. The natural gas network is linked to the power grid through the power to gas technology and the co-generation power plant, the last one also connecting the gas network with the high temperature district heating network.

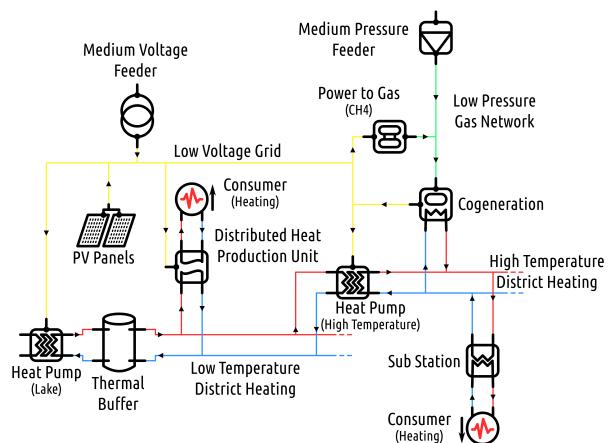


Figure 6: The simulated concept.

The main focus of this study is the presentation of the co-simulation framework as well as its application to the simulation based validation of a predefined design. The description below of the design methods and simulation models is needed to grasp the complete proposed approach. The following models and results are presented as a proof-of-concept of the proposed method.

### Distributed sub-systems

A selection of backbone consumers as been done based on various criteria: total heated surface, actual energy sources, year of construction, main buildings' usage, owners' situation (private or public), etc.

For those selected corresponding buildings, electrical, hot water production and internal heat gains profiles have been generated based on statistical values derived from data published by the Swiss Society of En-

gineers and Architects<sup>12</sup> (especially SIA2024). Statistical dispersion has been added to the produced profiles in order to avoid unrealistic correlation induced by using similar profiles.

Accurate building dynamic thermal models are quite complex therefore, a first order lumped parameter method was used to simulate the variations of the indoor temperature. In the lumped parameter method, each building was regarded as a whole from the point of view of a heating network and the indoor temperature of each building was an average value to transfer heat with the environment and the heat distribution system.

As shown by Zheng et al. (2018), the dynamics of the heat distribution system inside buildings have an influence on the performance of heat production systems. Circulation time of the water in the heat distribution system has been roughly estimated based on the total heated surface of the building.

The distributed heat production units (DHPU) are composed by a series of parallel heat-exchanger used as substations combined with heat pumps, a stratified thermal storage and dedicated controllers as shown in the Figure 7.

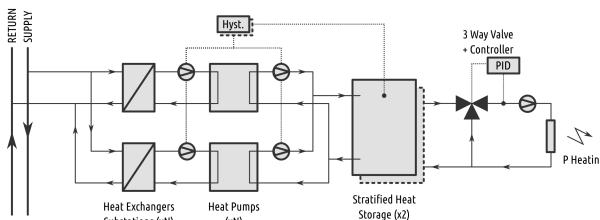


Figure 7: The simulated distributed heat production unit.

A rule-based design from engineering curated rule sets have been used to size each component of the distributed heat production units based on normative good practices defined in the Swiss Society of Engineers and Architects (especially SIA380/1)

The hot and cold outlet flow temperatures for the heat-exchanger used as sub-stations between the low temperature district heating network and the distributed heat pumps are estimated using the effectiveness-NTU method. The performances of the distributed heat pumps are computed as a static model with a piece-wise linear interpolation in 2 dimensions (cold and warm source temperatures) based on the corresponding operation data available in the manufacturer's catalogue<sup>13</sup> and updated at each simulation step. The series of heat pumps are controlled individually for each consumer using a cascade of hysteresis controllers combined with a set point for the

<sup>12</sup><http://www.sia.ch/en/the-sia/>

<sup>13</sup><https://www.hoval.ch>

heating supply temperature given by a heating curve based on the external temperature.

The behaviours of the stratified thermal storages used as thermal buffers are dynamically simulated using a Python implementation of a one dimension transient stratified water thermal storage tank model described in Rahman et al. (2016).

### Distribution sub-systems

The LT-DHN path has been chosen using a Python implementation (using library NetworkX described in Hagberg et al. (2008)) of an approximate resolution of the Steiner tree problem optimally linking the selected buildings identified as backbone consumers.

The Figure 8 presents the generated path of the LT-DHN as well as the connected buildings, the two HT-DHN areas of heat delivery and the location of the two heat pumps between LT-DHN and HT-DHN.

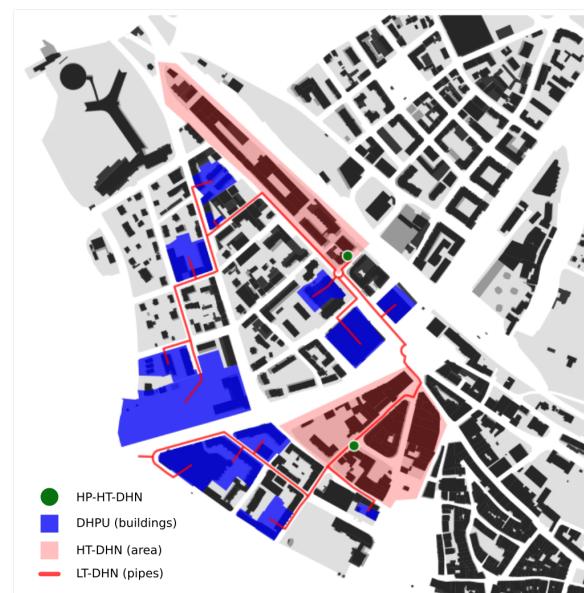


Figure 8: District heating networks concepts.

A Python implementation of the dynamic equation-based thermo-hydraulic pipe model proposed by van der Heijde et al. (2017) is used to simulate the behaviour of the two HT-DHN and the LT-DHN.

The medium and low voltage distribution power grids are modelled with PandaPower introduced by Thurner et al. (2017) and based on the real network of the city of Vevey. The studied area is supplied by a unique feeder from the high to medium voltage transformer, the simulation of the power grid is limited to this feeder.

The medium and low pressure natural gas networks are modelled using an isothermal, compressible gas flow model and are also based on the real gas infrastructure. In order to lower the complexity of the gas network model and reduce the computation time, the model has been simplified by (1) aggregating lined up

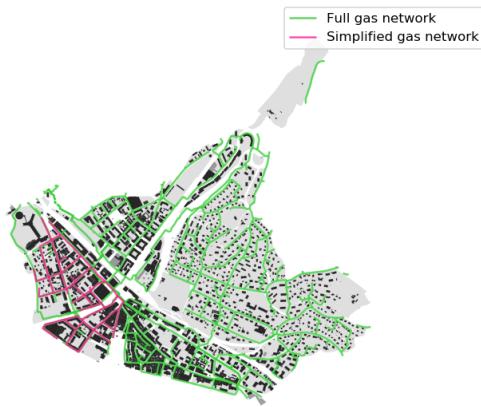


Figure 9: Full and simplified gas network.

pipes and (2) limiting the simulation to the part of the network in the studied area, using static mean gas flows based on measured values at the coupling points with the rest of the gas network (see Figure 9).

## Results

Figure 10 represents the loading of the power grid at the most stress-full time step of the first month of the 2007-2017 typical meteorological year, i.e. when the sum of the lines loading are maximum. It shows that the existing power grid is not able to provide enough energy to the proposed energy system concept without reaching limit values for the line loading and bus voltage variation.

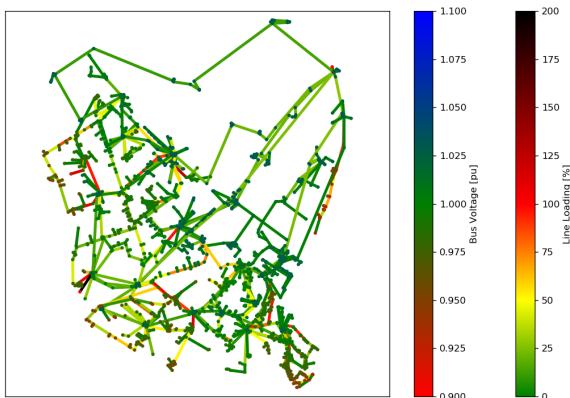


Figure 10: Power flow results for the grid.

This result has been produced by co-simulating all the components of the proposed energy system. All the components have been modelled using specific tools and different modelling paradigms, the co-simulation framework allowing to connect them together. Due to the large number of models run simultaneously, the simulation has been easily distributed over multiple computers to reduce the computation time without affecting the models.

## Discussion

The proposed approach allowed to assess the impact of a future concept of multi-energy system on the existing infrastructure. The precision of the global results depends essentially on the use and development of sufficiently accurate models of the simulated subsystems, an aspect which is not within the scope of this paper. The proposed models and concepts design are only used to demonstrate the use of the co-simulation framework and are therefore not the main contribution of this work.

The next step may be to assess the possibility of implementing multi-energy operation control strategies considering power grid limitation to tackle the grid over-loading problem. The test of such control strategies may impact the design of the proposed concept and therefore provide feedback to previous steps in the creation of a new concept for the energy system. Used in early steps of the planning and design process of a concept for an urban scale energy system, the precision of the presented method is highly limited by the data availability and uncertainty. Multiple iterations between concepts, design and validation steps may be necessary to provide solid and reliable results.

## Conclusion

The described tool provides an easy-to-use interface for initialising and executing technical simulations. It simplifies the communication with existing stand-alone simulators, with the aim to be as simple as possible for engineers from different fields to exchange information via the co-simulation. To that end, the co-simulation framework relies on state-of-the-art tools (Docker and ZeroMQ) for deployment and communication.

The dedicated conceptual and semantic layers as well as the deployment and results retrieval features have been integrated into a compact application programming interface delivered as a Python library. ZerOBNL is released as a stand-alone tool under the OSI approved Apache License 2.0 and the code is available online<sup>14</sup>. ZerOBNL is mainly based on ZeroMQ and Docker (Swarm and Compose). It has been developed in Python3.6 and tested on multiple operating systems.

## Acknowledgements

Research and implementation work presented in this paper has been carried out within the project IntegrCiTy<sup>15</sup> in the ERA-NET Cofund Smart Cities and Communities call. In Switzerland, it is funded by the Swiss Federal Office of Energy (contract SI/501404-2), as well as by the industrial and institutional partners of the project. Please visit the project website

<sup>14</sup><https://integricty.github.io/zerobnl/docu.html>

<sup>15</sup><http://iese.heig-vd.ch/projets/integricty>

for a full list of project consortium partners.

## References

- Eker, J., J. Janneck, E. Lee, Jie Liu, Xiaojun Liu, J. Ludvig, S. Neuendorffer, S. Sachs, and Yuhong Xiong (2003). Taming heterogeneity - the ptolemy approach. *Proceedings of the IEEE* 91(1), 127–144.
- Hagberg, A., P. Swart, and D. Chult (2008). Exploring network structure, dynamics, and function using NetworkX. In *Proceedings of the 7th Python in Science Conference*.
- Ilic, M. D., L. Xie, U. A. Khan, and J. M. F. Moura (2010). Modeling of future cyber-physical energy systems for distributed sensing and control. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 40(4), 825–838.
- Lilis, G., O. Van Cutsem, and M. Kayal (2018). A high-speed integrated building emulation engine based on discrete event simulation. *Journal of Systems Architecture*.
- Lund, H., S. Werner, R. Wiltshire, S. Svendsen, J. E. Thorsen, F. Hvelplund, and B. V. Mathiesen (2014). 4th generation district heating (4gdh). *Energy* 68, 1–11.
- Nghiem, T. X., A. Bitlislioglu, T. Gorecki, F. A. Qureshi, and C. N. Jones (2016). OpenBuildNet framework for distributed co-simulation of smart energy systems. In *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pp. 1–6. IEEE.
- Rahman, A., A. D. Smith, and N. Fumo (2016). Performance modeling and parametric study of a stratified water thermal storage tank. *Applied Thermal Engineering* 100, 668–679.
- Schutte, S., S. Scherfke, and M. Troschel (2011). Mo-saik: A framework for modular simulation of active components in smart grids. In *2011 IEEE First International Workshop on Smart Grid Modeling and Simulation (SGMS)*, pp. 55–60. IEEE.
- Thurner, L., A. Scheidler, F. Schäfer, J.-H. Menke, J. Dollichon, F. Meier, S. Meinecke, and M. Braun (2017). pandapower - an open source python tool for convenient modeling, analysis and optimization of electric power systems. *arXiv:1709.06743 [cs]*.
- van Beuzekom, I., M. Gibescu, and J. Slootweg (2015). A review of multi-energy system planning and optimization tools for sustainable urban development. In *2015 IEEE Eindhoven PowerTech*, pp. 1–7. IEEE.
- van der Heijde, B., M. Fuchs, C. Ribas Tugores, G. Schweiger, K. Sartor, D. Basciotti, D. Müller, C. Nytsch-Geusen, M. Wetter, and L. Helsen (2017). Dynamic equation-based thermo-hydraulic pipe model for district heating and cooling systems. *Energy Conversion and Management* 151, 158–169.
- Vogt, M., F. Marten, and M. Braun (2018). A survey and statistical analysis of smart grid co-simulations. *Applied Energy* 222, 67–78.
- Wetter, M. and P. Haves (2008). A modular building controls virtual test bed for the integrations of heterogeneous systems.
- Widl, E., T. Jacobs, D. Schwabeneder, S. Nicolas, D. Basciotti, S. Henein, T.-G. Noh, O. Terreros, A. Schuelke, and H. Auer (2018). Studying the potential of multi-carrier energy distribution grids: A holistic approach. *Energy* 153, 519–529.
- Widl, E., W. Muller, D. Basciotti, S. Henein, S. Hauer, and K. Eder (2015). Simulation of multi-domain energy systems based on the functional mock-up interface specification. In *2015 International Symposium on Smart Electric Distribution Systems and Technologies (EDST)*, pp. 510–515. IEEE.
- Zhang, Q., L. Liu, C. Pu, Q. Dou, L. Wu, and W. Zhou (2018). A comparative study of containers and virtual machines in big data environment. In *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, pp. 178–185. IEEE.
- Zheng, J., Z. Zhou, J. Zhao, and J. Wang (2018). Effects of the operation regulation modes of district heating system on an integrated heat and power dispatch system for wind power integration. *Applied Energy* 230, 1126–1139.