A Modular Co-Simulation Platform for Comparing Flexibility Solutions in District Heating Under Variable Operating Conditions

A Modular Co-Simulation Platform for Comparing Flexibility Solutions in District Heating Under Variable Operating Conditions

Pietro Rando Mazzarino, Martina Capone, Elisa Guelpa, Lorenzo Bottaccioli, *Member, IEEE*, Vittorio Verda, and Edoardo Patti

***Abstract*—Integrated modeling and simulation are crucial for optimizing cities' energy planning. Existing sector-specific analyses have implementation limitations in representing interactions across infrastructures, limiting optimization potentials. An integrated framework simulating multiple interacting components from a systemic perspective could reveal efficiency gains, flexibility, and synergies across urban energy networks to guide sustainable energy transitions. Co-simulation approaches are gaining attention for analyzing complex interconnected systems like District Heating (DH). Traditional single-discipline models present limitations in fully representing the interconnectivity between district heating networks and related subsystems, such as those in buildings and energy generation. Therefore, we propose a co-simulation based framework to simulate DH system behavior while easily integrating models of other subsystems and Functional Mock-up Unit (FMU) simulators. We tested this Plug&Play modular framework for Demand Side Management (DSM) and Storage-based strategies, evaluating their effectiveness in peak reduction while lowering the temperatures of the network.**

***Index Terms*—Co-simulation, demand side management, district heating, energy flexibility, peak shaving.**

I. Introduction

DISTRICT Heating (DH) is increasingly recognized as a key technology for the efficient and affordable decarbonization of the European energy system [1]. The ever-increasing role of DH in the transition to low-carbon heating strategies is linked to its ability to make efficient use of energy resources and facilitate the integration of renewable energy sources such as solar, geothermal, and waste-to-energy heat, as well as surplus heat from industrial plants and data centers [2]. Although well-established, DH systems are nowadays evolving into the fourth generation to exploit their full potential and meet the challenge of a future sustainable energy system [3]. Major innovations in the transition include the reduction of operating temperatures, which is essential for both renewable integration and reduction of heat losses [4], [5], [6], improvement of flexibility and storage systems [7], [8], [9], and smart integration of DH with other energy grids [10], [11].

In the field of district heating (DH) and multi-energy systems simulations, the integration of multi-domain approaches becomes very important to effectively address the growing complexity and heterogeneity inherent in these systems [12]. Co-simulation methodologies emerge as potent tools in managing the diverse domains present, finding extensive application within the Smart Grid context [13], [14], [15]. Consider, for instance, a scenario where we aim to simulate a district heating network together with various energy sources, different utility networks, and a multitude of prosumers. Using traditional single-discipline simulation environments to model the whole system might prove challenging. Herein lies the advantage of co-simulation: by seamlessly integrating multiple simulation tools and domains, it enables a more comprehensive representation of the entire DH system and the rest of the involved subsystems. Thus, in contrast to the monolithic nature of traditional simulation frameworks, co-simulation offers flexibility and modularity, enabling the incorporation of domain-specific models and simulation tools tailored to the unique characteristics of each subsystem. Moreover, co-simulation facilitates a more holistic analysis by considering the interdependencies and feedback loops between different components, thus providing insights into system behavior that may be overlooked in single-domain simulations. Leveraging co-simulation in scenarios characterized by small-scale or low complexity may not yield immediate performance enhancements, which, in our opinion, is an acceptable tradeoff considering: the inherent flexibility afforded by such architectures in scenario creation and the innate distributed and scalable nature of co-simulation tools, benefits that position co-simulation as an optimal solution for simulating the energy systems of the future.

To the best of our knowledge, the utilization of proper co-simulation frameworks for simulating District Heating (DH) or closely related energy systems is limited. The predominant approach in DH studies relies on monolithic, case-specific simulation frameworks. Upon scrutinizing studies that explicitly exploit the advantages of co-simulation, we have identified only three notable examples [16], [17], [18]. Nageler et al. [16]

Received 20 October 2023; revised 1 July 2024; accepted 22 August 2024. Date of publication 26 August 2024; date of current version 3 April 2025. This work was supported by the MUR-M4C2 1.5 of PNRR funded by European Union- NextGenerationEU through Project NODES under Grant ECS00000036. Recommended for acceptance by V. K. Prasanna. *(Corresponding author: Edoardo Patti.)*

The authors are with the Politecnico di Torino, 10129 Turin, Italy (e-mail: pietro.randomazzarino@polito.it; martina.capone@polito.it; elisa.guelpa@polito.it; lorenzo.bottaccioli@polito.it; vittorio.verda@polito.it; edoardo.patti@polito.it).

Digital Object Identifier 10.1109/TSUSC.2024.3449977

propose a simulation framework tailored for scalable scenarios involving DH models and buildings, albeit without considering storages. Vesaoja et al. [17] demonstrate the benefits of co-simulation in coupling control and physical aspects within DH networks; however, their work does not propose a generalized framework. Nikula et al. [18] present a co-simulation implementation for a dynamic process simulator and an event-based control system, tested on a small-scale case, which may not extrapolate well to larger applications. While these studies demonstrate the potential of co-simulation for DH systems, they also illustrate some gaps and limitations to address: i) Fully representing the complex behavior of DH networks and their interactions with storage requires more sophisticated co-simulation models. ii) Testing such models on real-world network topologies will provide more insights into their effectiveness for optimizing practical DH systems. To address these challenges, a more flexible co-simulation platform is needed. Thus, the main contribution of this work is to propose a flexible co-simulation framework for DH and to apply it to evaluate flexibility strategies. The approach proposed in this work addresses in a unique framework different levels of flexibility inherent in DH systems: the possibility of installing Thermal Energy Storages (TES) and the implementation of Demand-Side Management (DSM) are taken into account as two effective solutions to reduce the thermal peak arising in the morning due to the attenuation/shutdown of the heating systems during the night, which is typical in Mediterranean regions and negatively affects the performance of the entire production system. After having shown in [19] that these two strategies can be adopted simultaneously as complementary solutions (and possibly combined with further flexibility solutions provided by multi-energy systems) to improve the operation of DH and of the global energy system as well, in this paper we evaluate their changing influence under different scenarios of supply temperature reduction. To summarize, the novel contributions of this work can be grouped into two macro areas: i) novelty in the cosimulation platform and ii) novelty in the application scenario.

Regarding the former contribution we are proposing:

- a modular framework integrating heterogeneous models as Functional Mock-up Units (FMUs) or code through well-defined interfaces. The interfaces are tailored to the specific needs of simulating typical District Heating components. This contribution consists of major modifications of our previous work [20];
- a generalized FMU adapter enabling integration of diverse component models.

Regarding the latter contribution to the application scenarios, the scientific novelty can be summarized:

- the evaluation of storage-based and demand-side management (DSM) strategies on a realistic network;
- the evaluation of the impacts of supply temperature reduction on storage-based and DSM peak shaving strategies through simulations on a representative district heating network case study.

The paper is structured as follows. Section II describes the proposed methodology and the framework structure, analyzing its layers and components. Sections III and IV present the case study used to test the framework and the experimental results of the performed simulations, respectively. Finally, Section V provides concluding remarks.

II. Co-Simulation Framework for Energetic Analyses in DH

The proposed co-simulation framework enables various energetic analyses in district heating systems. It exploits MOSAIK as co-simulation engine. Thus, our framework provides the following main features:

- Plug&Play capability by integrating heterogeneous models flexibly;
- Modularity through decomposing the system into independent simulators;
- Scalability in simulating large grids by coupling many models efficiently;
- Ability in coupling models with different characteristics;
- Distribution of computational effort across multiple computer systems.

Fig. 1 shows the framework architecture, which consists of two main macro-layers i) the *Data Source Layer* and ii) the *Co-simulation Layer*. The *Data Source Layer* includes all the inputs and the tools for Urban energy modelling needed by the upper layers. The *Co-simulation Layer* encompasses the main co-simulation components: i) the models to simulate physical behaviours, ii) the interfaces to execute models during simulation, and iii) the orchestration engine to synchronize the interfaces among them. The following sections will describe in-depth the whole framework.

A. Data Source Layer

The *Data source Layer* is responsible for managing physical information about the system. It exposes two main functionalities: static databases for *Real data inputs* and a tool for *Synthetic grid creation* (see Fig. 1).

In our system, the *Data source Layer* manages physical information, offering static databases for *Real data inputs* and a tool for *Synthetic grid creation* (see Fig. 1). *Real data inputs* encompass *District heating topology*, *Thermal Power demand*, *Mass flow demand*, and *Weather* data. If *District heating topology* is absent, it can be generated by the *Synthetic Grid creation* tool. Fig. 2 illustrates a simple example for the *District heating topology*, detailing system actors (*Thermal power plant*, *Thermal storage*, and *Buildings*) and the network's structure. The DH network is segmented into delivery and return sub-networks, both depicted as directed graphs where pipes are represented as edges and junctions or interface nodes as nodes (see Fig. 2). Delivery and return sub-networks are topologically identical but with opposite edge directions (red and blue arrows in Fig. 2). Real-world DH networks form closed circuits, making interface nodes the connection points between delivery and return sub-networks. Employing a graph data structure, attributes such as pipe lengths and diameters are assigned to edges, while nodes store actor information (*Thermal power plant*, *Thermal storage*, or *Building*), if applicable.

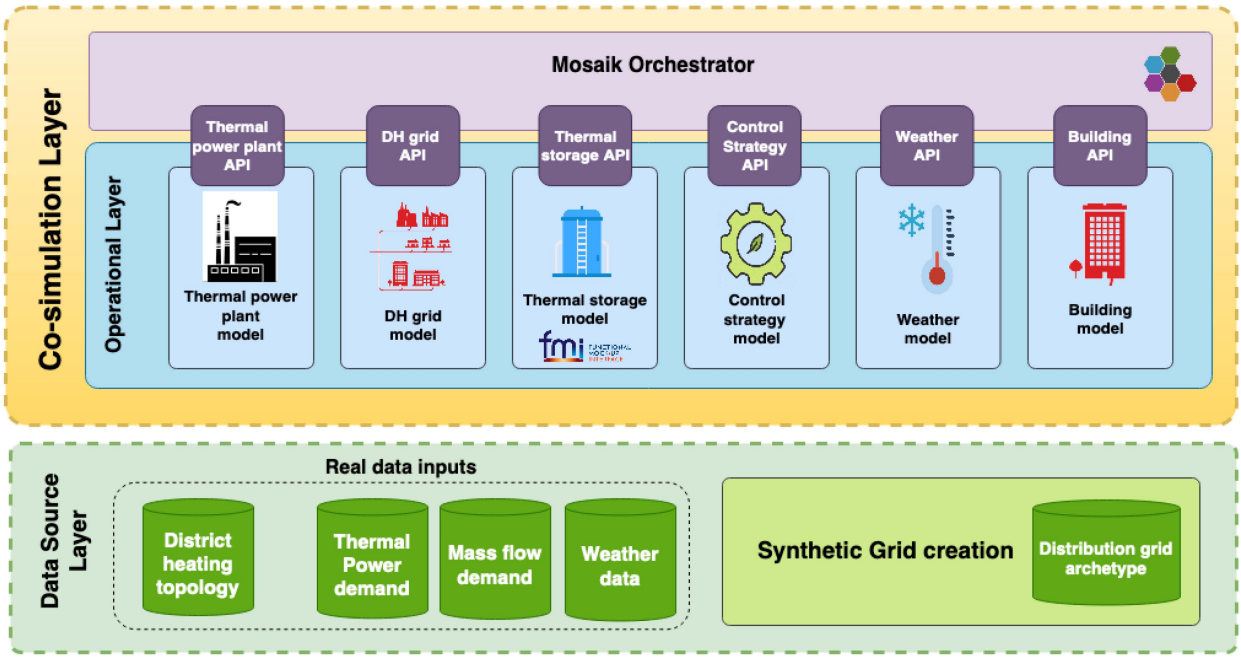

Fig. 1. Platform schema.

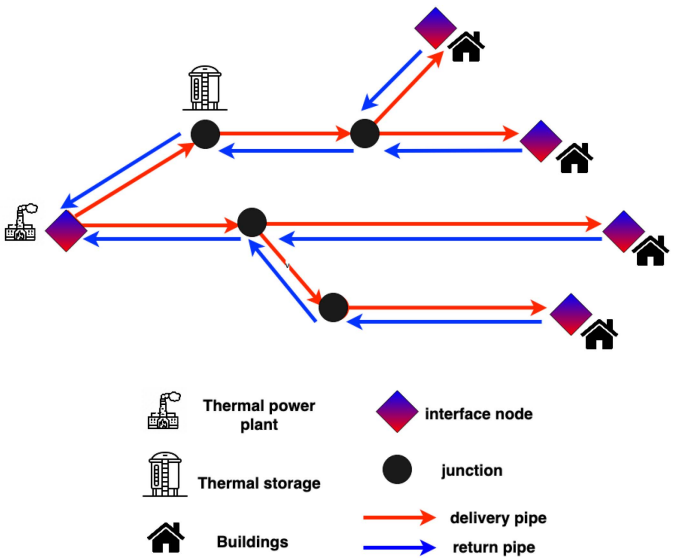

Fig. 2. Example of a DH network.

Thermal Power demand, *Mass flow demand*, and *Weather data* constitute time-series information (see Fig. 1). They represent the thermal power demand from buildings, associated mass-flow, and outdoor temperature throughout the simulation period. These inputs can either be static within the *Data source Layer* or dynamically generated by dedicated simulators in the *Co-simulation Layer*. The *Synthetic Grid creation* tool is employed when no *District heating topology* (see Fig. 1) is provided, necessitating a synthetic one. It utilizes a real-world *Distribution grid archetype* as a basis to construct realistic and intricate heating networks. Multiple replicas of this archetype grid are interconnected to form new topologies, with configurable parameters such as the number of replicas, interconnection nodes, and locations of *Thermal power plants*, *Buildings*, and *Thermal storage*. Pipe design in the new grids involves determining diameters and lengths to achieve a target fluid velocity of approximately 2 m/s based on nominal demand values. This modular and scalable grid generation approach relies on composing archetype replicas and dimensioning pipes parametrically based on hydraulic design principles. The *Data source Layer* facilitates high configurability, allowing various usage possibilities. Real data or simulators can be utilized. The district heating network topology and components (number of thermal power plants, storage units, consumers, transport and distribution grids) can either leverage an external case study or be generated by using an editable configuration file and a sample of a real distribution grid. This dual approach confers versatility and broad applicability of the co-simulation framework.

B. Co-Simulation Layer

The *Co-simulation Layer* is the core of our framework and, as shown in Fig. 1, it consists of the *Operational Layer*, the MOSAIK APIs, and the *Mosaik Orchestrator*. This configuration is the key enabler for the high modularity and Plug&Play capabilities of the framework. Indeed, the *Mosaik Orchestrator*, along with the MOSAIK APIs, manages only the co-simulation operations (i.e. Synchronization, data-exchange, and instantiation), giving high flexibility to the layer below, the *Operational Layer*.

The *Operational Layer* includes physical and control models of the real-world entities to be simulated (i.e. *Thermal power plant model*, *DH grid model*, *Thermal storage model*, the *Control strategy model*, *Weather model* and *Building model*). The models

in this layer can be easily extended, changed, and replaced because they are separate entities combined by MOSAIK APIs. Indeed, they only need to meet the specific input/output requirements specified in the APIs meta description. The models can be Python software or a Functional Mock-Up Unit (FMU). The FMU standard allows integrating models written in any modeling language or platform compliant with this standard. The models in the *Operational Layer* simulate the main components of a generic DH system, and they are reported in Fig. 1 and described in the following. In this platform we have mainly prepared standardized interfaces for the most important DH components, then to test them we have encapsulated physical models as well as real-world data readers or compact models. Indeed, regarding *Building model* and *Weather model*, they are simple data readers, outputting respectively, the external temperature and the building power demand. For what concerns the *Thermal power plant model*, it encapsulates a Python model that performs mass balance calculations to calculate the mass-flow rate to be supplied to the network. It is worth noting that in line with the main concept of this framework, substituting data readers or simple models with more complex models is easy and configurable.

The *DH grid model* is a one-dimensional quasi-steady state model as in [20]. This is a common approach in district heating applications since pressure perturbations travel within the network at sound velocity; therefore the time pressure wave, taken to reach the furthest areas, is much smaller than the time step considered in the simulations. The same does not apply to the temperature that travels at the velocity of water that is small (usually 0.5–4 m/s). In large networks, the water flow can take up to an hour or more to reach the furthest areas of the network. For an in-depth description of the model, we refer to [21] and in the following, we report the set of the matrix equations used: Equation 1 for computing the mass, Equation 2 for the momentum and Equation 3 for the energy. The model has been implemented in Python and takes as inputs the injected mass-flows and the temperature at the injection nodes.

The *Thermal storage model*, represents a buffer water storage stratified into a configurable number of layers that are connected to allow heat and flux transfer. The top layer is coupled with return colder water while the bottom one with the supply warmer water. The heat transfer between the layers can be modeled by considering the conductance or by also including the buoyancy effects through the AixLib implemented heat transfer model: HeatTransferBuoyancyWetter. The original model has been taken from AIXLIB [22], thus its source code is written in MODELICA language, and it has been enhanced to be used for simulating both charge and discharge as well as to be exported in FMU standard. Fig. 3 reports all the variables of interest exported by FMU. These are considered as inputs and outputs of the model and are the only exchanged variable during a co-simulation. The modifications only regard the interfaces for data communication without affecting the validity of the underlying physical model from AIxLib. The inputs are explained in the following. The *Charging temperature* and the *Charging mass-flow* fix the temperature and the mass-flow when the thermal storage is charging. The *Discharging temperature* and the *Discharging mass-flow* are

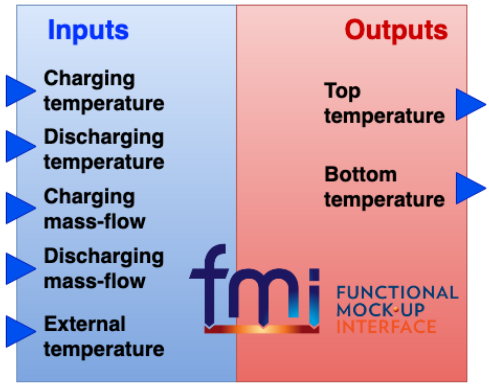

Fig. 3. Thermal Storage FMU I/O points.

the equivalent values for the discharging phase. These inputs are used only when the *Thermal storage model* is in one of the two respective states (charging/discharging). For example, if it is in charging state, it will receive and use the values for *Charging temperature* and *Charging mass-flow* while setting to zero those related to the discharge phase. The only input that is sent to the *Thermal storage model*, regardless of the operating state, is the *External temperature* which is needed to calculate heat losses through the ambient. On the other side, the outputs are the temperature values of the bottom layer and the top layer of the water tank, namely the *Top temperature* and the *Bottom temperature* in Fig. 3. These are the two needed outputs for describing the outflow of water from the storage tank. Indeed, the valves that connect the thermal storage to the heating network are located at the top and at the bottom of the water tank. During co-simulation, the only visible variables are the inputs and the outputs, nevertheless during the initialization of the model it is possible to configure a series of parameters that specify both geometrical and thermal properties, such as height and diameter of the tank, thermal transmission properties of the material used and number of layers.

$$A \cdot G + G_{ext} = 0 \tag{1}$$

$$G = Y \cdot A^T \cdot P \tag{2}$$

$$M \cdot \dot{T} + K \cdot T = g \tag{3}$$

where:

A = incidence matrix;
A^T = transpose incidence matrix
G = vector of mass flow rate within each branch;
G_{ext} = vector of the mass flow rate entering and exiting each node;
Y = conductance matrix (inverse of the fluid dynamic resistance);
P = vector of pressures in the nodes;
M = mass matrix;
K = stiffness matrix;
T = vector of temperatures in nodes;
g = known-term vector of the equation.

The *Control Strategy Model* module oversees the implementation of system management and control within the framework.

This Python-based module orchestrates strategies for peak reduction, comprising two distinct approaches. The initial method capitalizes on thermal storage, while the alternative hinges on the dynamic shifting of combined building load profiles (DSM). Operationally, the former generates schedules for thermal storage, delineating periods of charging, discharging, or idle states. It also governs the rate of mass-flow entering or exiting the thermal storage, maintaining a constant flow for charging and a flow rate proportional to the morning peak's slope for discharging. While optimal discharge scheduling is beyond the scope of this study, the platform's modular design facilitates seamless integration of optimal strategies in a Plug & Play manner. Conversely, in scenarios where the DSM strategy is invoked, the *Control Strategy Model* restructures the thermal power and mass-flow demand inputs from buildings to preemptively manage loads and mitigate aggregated demand peaks. A detailed analysis of the sub-optimal shifting mechanism is available in [23]. It's imperative to note that for transparency, the DSM strategy necessitates the provision of complete time-series data for thermal power and mass-flow demands at the simulation's onset. To enable runtime utilization of the DSM strategy, where demand data are generated dynamically by simulators (as detailed in Section II-A), a distinct *Control Strategy Model* configuration is required.

The *Mosaik Orchestrator* and the MOSAIK APIs manage the pure ICT aspects of co-simulation i.e. the Synchronization and the data-exchange. Their implementation relies on the MOSAIK python library, which offers the orchestrator engine and a high-level APIs base class. In line with the definitions in MOSAIK documentation [24], the APIs are called simulators. The simulators instantiate the models and it is possible to have several model and simulator instances. Our framework provides a simulator for each model in the *Operational Layer*, thus: i) *Thermal power plant API*, ii) *DH grid API*, iii) *Thermal storage API*, iv) *Control strategy API*, v) *Weather API* and vi) *Building API* (see Fig. 1). The simulators create a number of instances of the models that are equal to the number of entities to simulate. For example, if we have to simulate ten buildings, there will be a unique *Building API*, which will instantiate ten instances of the *Building model*. Each instance will have a unique id and will be reported to the *Mosaik Orchestrator*, during the creation phase. Hence, the simulators are implemented as pure interfaces between the model instances and the co-simulation process. The simulators aggregate outputs and distribute inputs from and to the model instances. The design is the same for all the simulators except for the *Thermal storage API*, which is the only one interfacing with an FMU model. Due to this exception, a particular effort has been done to standardize a Mosaik API for FMU models, resulting into two different standards of simulators, for Python models and for FMU models.

Finally, the role of *Mosaik Orchestrator* is to manage the simulation process by connecting and timing all the simulators. Its main role is to act as an intermediary for the data exchange between the simulators. Similarly, the simulators manage inputs and outputs for their model instances. The *Mosaik Orchestrator* keeps track of all the relationships between simulators and model instances in the simulated environment. It routes the data-flow thanks to a dictionary data structure in which is kept the knowledge of time, connections, and data to be exchanged. Thanks to this knowledge it distributes information to each simulator and finally allows them to advance through time.

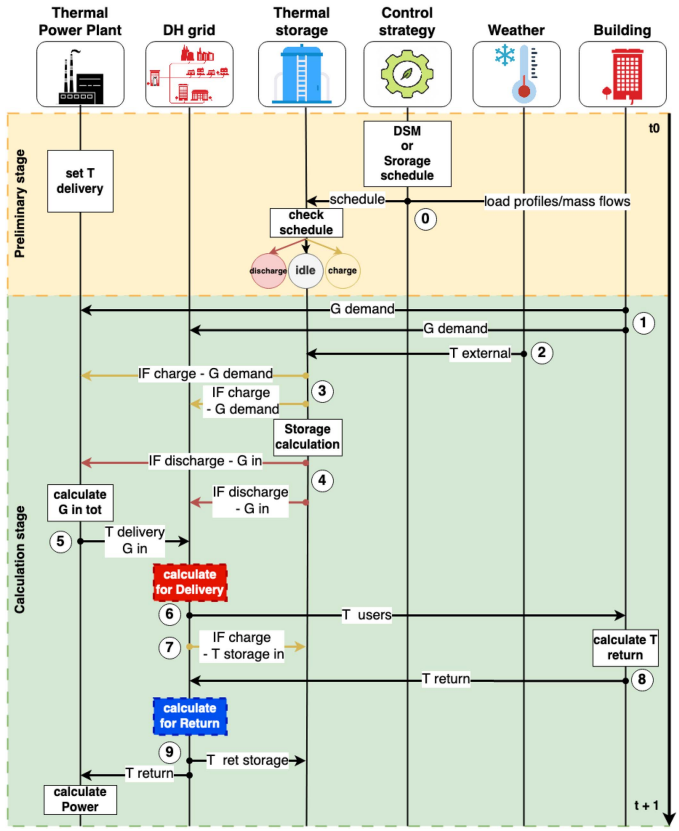

Fig. 4. Schema of the co-simulation workflow.

C. Co-Simulation Workflow

This section presents the communication workflow of all the simulators involved in our solution. As shown in Fig. 4 they are: *Thermal power plant*, *DH grid*, *Thermal storage*, *Control strategy*, *Weather*, and *Building*. Simulators can have multiple instances and instantiate one model multiple times, for simplicity the image reports only the general data exchange workflow between simulators typologies. The simulation process includes all time-based simulators that interact through a cyclic data-flow [24]. In a single time-step, each simulator can receive and send information to multiple others before or after having performed its own calculation.

Fig. 4 schematizes the co-simulation workflow for a single time-step. The blocks on the temporal lines for each simulator represent the simulator task, while the numbered arrows represent the performed data-exchange. The workflow is divided into a *Preliminary stage* and a *Calculation stage* as described in the following.

Preliminary stage: Control strategy communicates either the schedule to *Thermal storage* or the anticipated load profile to *Building*, according to the chosen strategy. *Thermal storage*

checks if it is in idle or in charge/discharge state. *Thermal power plant* fixes the supply temperature.

Calculation stage: The *Building* sends the mass-flow demand to both *DH grid* and *Thermal power plant* to perform the mass balance (see ① in Fig. 4). *Weather* sends the outdoor temperature to *Thermal storage* (see *T external* and ② in Fig. 4). *Thermal storage* performs the following consecutive tasks: i) it sends its mass flow demand to both *Thermal power plant* and *DH grid*, if it is in charging state (see ③ in Fig. 4); ii) it performs its core computation as described in the previous Section II-B, which is represented by the *Storage calculation* box in Fig. 4; iii) it communicates the discharging mass-flow to both *Thermal power plant* and *DH grid*, if it is in discharging state (see ④ in Fig. 4). Once all the information required is received, the *Thermal power plant* calculates the mass flow to be injected into the network and it sends this result to *DH grid* (see *calculate G in tot* box and ⑤ in Fig. 4). *DH grid*, receiving all its inputs, computes the thermo-fluid dynamics for the delivery (see *calculate for Delivery* box). The results of this calculation are all the temperatures of both junctions and nodes in the delivery network. *DH grid* sends the proper temperature values to both *Building* and *Thermal storage* (see *T users*, ⑥, *T storage in*, ⑦ in Fig. 4).

Building can now calculate the temperature of the water after the thermal power extraction (see *Calculate T return*), it injects the water into the return network and it sends these values to *DH grid* (see ⑧ in Fig. 4). *DH grid* performs the thermo-fluid dynamics calculation for the return sub-network and provides the resulting temperatures to *Thermal power plant* and *Thermal storage* (see ⑨ in Fig. 4). Finally, given the fixed supply temperature and the incoming temperature of the return network, *Thermal power plant* computes the final thermal power to heat up the water that will be injected into the delivery sub-network in the next simulation time-step.

III. Experimental Setup

The ability of our framework and models to scale up and create synthetic grids has been presented in [20]. Thus, in this work, we focus on showing modular flexibility by integrating new models and performing a comparative case study analysis. In this Section, the case study is explained. Thus, the proposed analysis focuses on peak reduction strategies in the context of supply temperature reduction. Testing strategies, integrating models, and shifting the analysis perspective are all features enabled by the co-simulation platform. Indeed, we can assess different peak reduction strategies (as well as general flexibility strategies) and simulate them simultaneously as well as separately. The second case is the one we implement in this paper which allows us to propose a fair comparison between different strategies as well as analyze the effects of each one on the different components of a DH infrastructure and under varying boundary conditions (e.g. the supply temperature reduction).

The simulated environment is a distribution grid, supplying pressurized water and located in a city in Northern Italy. The network topology as well as thermal power demand and mass flow data belongs to this real-world case scenario. The DH delivery

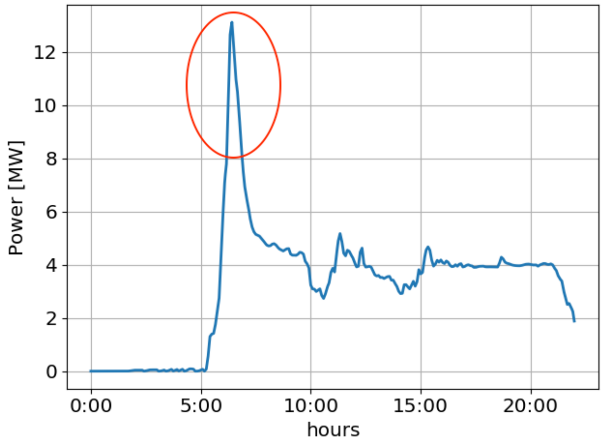

Fig. 5. Load profile of the given network at the Thermal power plant node.

sub-network is a radial network without loops composed of 202 nodes and 201 edges. It supplies 47 buildings and is populated with one thermal storage with a capacity of around 26150 liters and one thermal power plant. In this way, the analyzed grid portion is isolated from the transport grid for which we have no data. The simulations are performed using demand data and weather conditions of a typical winter day of consumption, the demand data are well representative of the problem we want to analyze and for which we tested the peak reduction strategies. In particular, thermal power demand and mass flow data come from real-world measurements for what concern the operations at 120 °C supply temperature, while for the other scenarios at lower temperatures, they have been taken from [21].

Fig. 5 shows the aggregated load profile of the chosen case study at the thermal power plant. This graph clearly reports an early-morning peak, approximately between 6:00 and 7:00 a.m. This depends, especially in less harsh areas, on heating devices being switched off (or attenuated) during night hours. Indeed, the used data are mainly representative of residential central heating systems which, by the Italian custom, are switched off during the night. From the perspective of smart energy management, load peaks must be avoided to reduce prices and keep the thermal power level of production plants as constant as possible. A straightforward method for reducing peaks is integrating thermal storage systems. Another possible solution involves acting on the demand load profiles by shifting some of the users' consumption peaks to other time slots.

Accordingly, we propose two different peak reduction strategies, which are implemented as two alternatives in the *Control strategy model* in Fig. 1. The first one is based on the usage of thermal storage which is scheduled according to the needs of the load profile of the case study. The second one, instead, exploits Demand Side Management (DSM), thus it modifies the building's load profiles by shifting them in time, particularly advancing the demand of some minutes. The analysis focuses on understanding the behavior of both these strategies under the effects of reducing the supply temperature in the district heating network. As the supply temperature decreases, the performance of the peak reduction strategies is investigated to reveal how

TABLE I
NUMBER OF BUILDINGS AND THE RELATIVE THERMAL POWER DEMAND ADVANCE IN MINUTES

number of buildings	time advance [min.]
35	30
3	25
4	20
1	5
4	not shifted

TABLE II
RESULTS OF PEAK REDUCTION FOR BOTH DSM AND STORAGE STRATEGIES FOR EACH SUPPLY TEMPERATURE

Supply temperatures [°C]	ΔP Storage [%]	ΔP DSM [%]
120	18.0	33.3
110	17.2	33.5
100	16.8	34.5
90	14.9	33.4

the thermal storage strategy is impacted compared to the DSM strategy and how sensitive each approach is. Comparing the absolute effectiveness of the strategies is beyond the scope of this work and would depend on several implementation details. The aim is to gain insights into how each approach responds differently to lower supply temperatures.

The thermal storage control strategy follows a semi-optimized schedule: It charges the storage from 3:00 to 5:20 with a fixed mass flow rate of around 3.094 kg/s and discharges during the peak hour (6:00 to 7:00) with a variable mass flow rate that follows the steep increase in demand during that hour. By charging overnight when demand is low and discharging during the peak hour, the storage helps flatten the load and reduce the peak demand placed on the grid.

On the other side, for the DSM based strategy, we anticipated some buildings' load profiles, resulting in an aggregated load curve with a reduced morning peak. Table I shows the number of buildings that experienced a load shift and the amount of shift in minutes. A majority of buildings saw a 30 min advance in their load, while some saw 20–25 min shifts, only one was shifted by 5 minutes, and four experienced no shifting. This configuration, as well as the storage schedules, are not mathematically optimized but chosen based on the authors' experience and qualitative assessment, especially for what concerns the DSM shiftings which in reality are strictly dependent on users' preferences and thus subjected to high uncertainty.

In summary, the simulations aim to study the effects of both the storage and the demand side management (DSM) strategies with a constant decrease in the inlet temperature. The simulations were performed as follows:

- A base simulation with no peak reduction strategy;
- A simulation using the DSM strategy;
- A simulation using the storage strategy.

This process was repeated for each of the supply temperature values (120/110/100/90 °C), allowing comparisons of the base case, DSM strategy, and storage strategy to be made at each temperature.

IV. EXPERIMENTAL RESULTS

In this section, we present the results of the performed simulations. Fig. 6 shows the resulting load profiles for each simulation, for better visualization only snapshots focused on the morning period (3:00–9:00) are presented. The resulting load profiles are the thermal power that is withdrawn from the thermal power plant plus the network losses. The reported four graphs are one for each tested supply temperature Fig. 6(a) 120 °C, Fig. 6(b) 110 °C, Fig. 6(c) 100 °C, Fig. 6(d) 90 °C, and the line color schema is the same for all. The solid blue line reports the thermal power curve without any peak reduction strategy, the dot-dashed green line the thermal power curve under the DSM strategy, and the dashed orange line the thermal power curve under the storage-based strategy.

Fig. 6 shows that both DSM and storage strategies flatten the peak to some extent. This is visible in the mismatch of the curves under the peak of the baseline. In order to compensate for the peak reduction between 6:00 and 7:00 both the DSM and the storage profiles consume more than the baseline in other periods. For both strategies, these periods are visible before the peak, indeed the storage is charged during the night and the DSM has anticipated the loads. Comparing the absolute effectiveness of the two strategies is beyond the scope of this work and would depend on implementation details. However, the results show that as supply temperature decreases, the DSM strategy is less affected in terms of peak reduction compared to the storage-based strategy. In summary, for the same DSM and storage configurations, reducing supply temperatures has a greater impact on the performance of the storage strategy in flattening load peaks compared to the DSM strategy.

To better understand the results from a quantitative perspective, we have reported in Table II the peak reduction percentages for both strategies for each simulated supply temperature. The chosen index to evaluate the effectiveness of the peak reduction is the *Relative Peak thermal power demand reduction* taken from [25] and reported in Equation 4, where P_{flex} represents the peak thermal power when a flexibility strategy is applied and P_{max} the original thermal power peak of the baseline.

$$\Delta P\% = 1 - \frac{P_{flex}}{P_{max}} \tag{4}$$

The results in Table II indicate that: The storage solution becomes less effective at reducing the peak demand as the inlet temperature decreases. The percentage of peak reduction achieved by the storage strategy drops from 18.04% at 120 °C inlet temperature to 14.95% at 90 °C. This suggests that lower inlet temperatures reduce the effectiveness of the storage in flattening load peaks. In contrast, the DSM solution is able to maintain a more constant level of peak reduction, ranging from around 33% to 34% across the different inlet temperatures. Indeed, the relationship between DSM strategy effectiveness and the lowering of supply temperatures is not linear, small casual

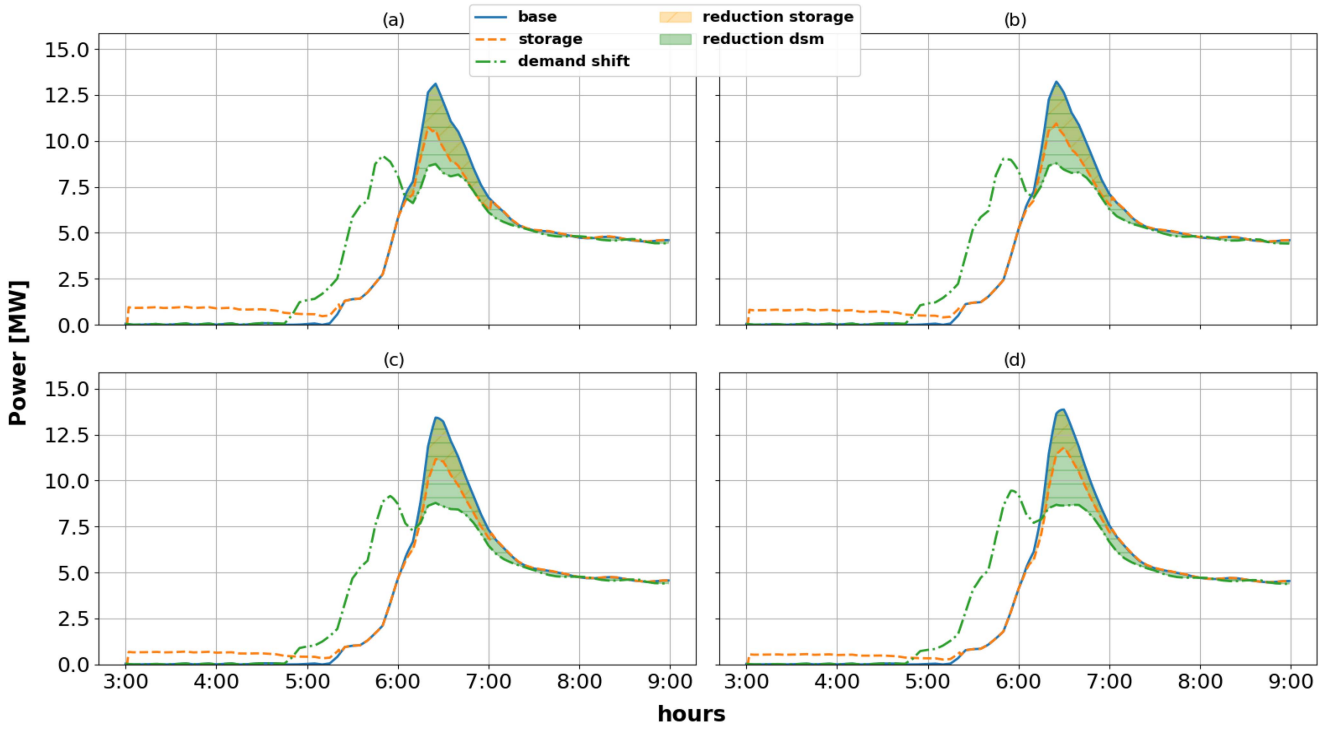

Fig. 6. Snapshots of the withdrawn thermal power from the Thermal Power plant for the performed simulations, focusing on the morning peak. (a) case with supply temperature 120 °C (b) case with supply temperature 110 °C (c) case with supply temperature 100 °C (d) case with supply temperature 90 °C.

oscillations are present. Nevertheless, the maximum variation in performance is around 1.22 percentage points which can be seen as a negligible improvement leading to consider the DSM performance rather constant. This implies that shifting and adjusting users' demand loads can achieve similar levels of peak reduction, regardless of changes in the inlet supply temperature. The trends in the data, therefore, show that DSM presents a solution for reducing peaks that is less sensitive to variations of the inlet temperature, while the impacts of storage are more dependent on the temperature of the water entering the distribution grid.

A. *Thermal Storage Results*

Co-simulation allows us to perform different analyses by coupling simulators. We can choose which perspective and attributes to examine. In this section, we present the evolution of thermal storage temperatures during the simulation. Fig. 7 shows the temperatures of the top (*T top*) and bottom (*T bottom*) layers of the storage along with the average temperature (*T avg*). The graph is extracted for the simulation with 90° C of supply temperature. The two highlighted areas correspond to charging and discharging periods. During charging, all temperatures increase at different rates due to their positions and proximity to the inlet valve.

During discharging, we see a drastic decrease in the bottom temperature. Indeed, during the discharge phase, the water leaves the storage from the top layer and colder water enters from the bottom one. The cold water entering comes from the return sub-network, which is much colder than the water in the storage,

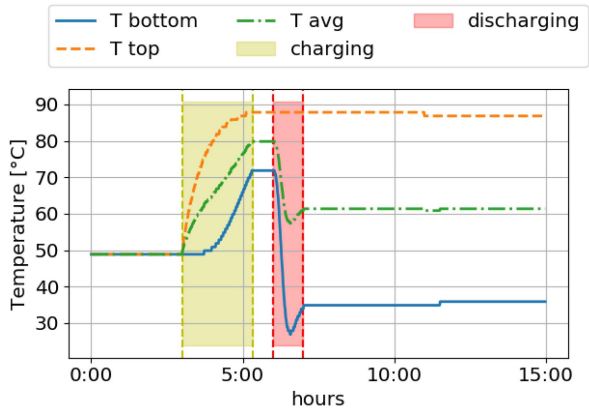

Fig. 7. Temperature profiles of upper, middle and lower layers in the thermal storage during charge/discharge phases.

causing the temperature to drop. Whilst, the top temperature is only slightly affected, being the temperature of the last layer before the outlet valve for hot water. In idle periods the curves slightly converge, illustrating the effects of mixing and heat losses (see from the end of discharging period to 15:00). In particular, we can see that in idle periods the temperature changes are small meaning that mixing is slow and also heat losses are small. The first aspect depends on the fact that there is a very slow fluid exchange between layers. Whilst, the second aspect depends on the high insulating properties of the chosen tank and on the not extreme outdoor temperatures of the used weather file, given as input to the simulation.

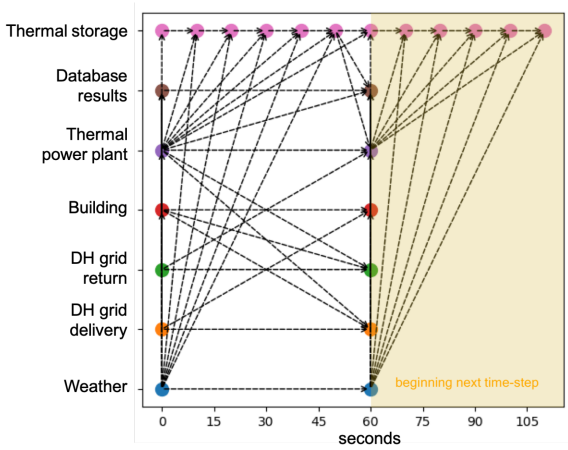

Fig. 8. Co-simulation execution graph for 2 time-steps.

B. Execution Graph

In this section, we address the resulting data-exchange workflow of the simulated entities. The execution graph in Fig. 8 reports each simulator state over time and their interactions. The colored dots represent simulator entities states and the arrows represent data exchanged between entities. The graph shows a 60 s time step and the beginning of the next one, with time on the x-axis and entity names on the y-axis. This graph is the real implementation of the data workflow previously explained in Section II-C. The involved simulators are shown on the y-axis in Fig. 8. The *DH grid* simulator has been divided into *DH grid return* and *DH grid delivery* only for better visualization. The *Database results* simply collects the outputs of all the simulators every time step to store them in hdf5 databases. We never cited it due to its marginality, but it is reported in this execution graph because it still takes part in the actual data-exchange process by receiving data every time-step. Then, besides these two differences, the other simulators are the same as described in Section II. Fig. 8 shows that each simulator advances through time sending and receiving data. There are two types of data-exchange between simulators: i) instantaneous and ii) time-shifted. The first kind of exchange is the one represented by the vertical arrows between simulator states (e.g. at $t = 0$), in this case, the data are sent between each simulator following a defined order at the beginning of the time-step execution. The second kind of exchange is represented by the inclined arrows that connect states at different times (from which *"time-shifted"*). The time-shifted exchange is a common way to resolve cyclic data-flows (see MOSAIK documentation [24]) in the co-simulation domain. Indeed, it is needed when two simulators need to perform multiple data-exchange in the same time-step. Shifting in time these messages is an approximation that can be resolved with some initialization to obtain the correct alignment between simulators. In conclusion, it is worth noticing that the *Thermal storage* simulators presents six states in a single time-steps. Indeed, due to its own model, the *Thermal storage* simulator has a different time resolution with respect to the other simulators, it is stepped every 10 seconds while the rest of the involved simulators every 1 minute. This shows the flexibility of the platform that allows to couple simulators with different time-resolutions.

V. CONCLUSION

The co-simulation framework demonstrates the potential to optimize district heating systems through a holistic view that integrates models from different domains. By coupling simulators in a flexible Plug&Play fashion, the framework shows it can simulate realistic scenarios with various perspectives. In the context of our manuscript, it is important to acknowledge that a performance comparison of our co-simulation framework with traditional platforms like Modelica may not be entirely fair. These platforms serve distinct purposes and offer different capabilities. Our framework offers the possibility of integrating diverse simulation tools (e.g. Modelica, Eplus, Simulink etc.), and programming languages, in particular custom Python models. This approach supports the implementation of complex control strategies on top of diverse systems with increased levels of detail, complexity, and scalability. Achieving this on traditional monolithic platforms (e.g. Modelica or Matlab) can be challenging especially when the need for multidisciplinarity, scalability, and complexity increases. Therefore, rather than conducting direct comparisons with traditional platforms, our manuscript emphasizes the unique advantages of our co-simulation approach by showing its implementation and usage on a DH system. The simulations conducted to compare demand-side management and storage-based peak reduction strategies reveal that DSM performs more reliably under lower supply temperatures. This is crucial for the future of District heating networks which, to be more efficient and to integrate renewable technologies, will inevitably lower their supply temperatures. Indeed, understanding the limits of peak reduction strategies for future DH scenarios is important to direct research efforts in the right direction, in this regard, DSM strategies seem very promising. Interesting future analyses could include models for optimal thermal storage scheduling as well as optimal DSM shiftings, in addition focusing on DSM will need the integration of some consideration on people's behaviors and choices or on new contractual strategies to actually verify the feasibility of the shiftings. To conclude, the platform's modularity enables easier benchmarking and exploration of opportunities not only for District heating systems but more in general for what will be our main focus: multi-energy system analyses. This demonstrates its potential usefulness for studying complex urban energy systems and informing efficient solutions. With further development and refinement, this co-simulation approach holds promise for identifying flexibility options and optimal operating strategies to minimize costs and emissions for future sustainable heating networks that exploit synergies across infrastructures.

REFERENCES

[1] H. Lund et al., "4th generation district heating (4GDH): Integrating smart thermal grids into future sustainable energy systems," *Energy*, vol. 68, pp. 1–11, 2014.

[2] A. Jodeiri, M. Goldsworthy, S. Buffa, and M. Cozzini, "Role of sustainable heat sources in transition towards fourth generation district heating–a review," *Renewable Sustain. Energy Rev.*, vol. 158, 2022, Art. no. 112156.

[3] H. Lund et al., "The status of 4th generation district heating: Research and results," *Energy*, vol. 164, pp. 147–159, 2018.
[4] M. Rämä and K. Sipilä, "Transition to low temperature distribution in existing systems," *Energy Procedia*, vol. 116, pp. 58–68, 2017.
[5] P. Sorknæs et al., "The benefits of 4th generation district heating in a 100% renewable energy system," *Energy*, vol. 213, 2020, Art. no. 119030.
[6] A. Dalla Rosa et al., "IEA DHC Annex X report: Toward 4th generation district heating: Experience and potential of low-temperature district heating," pp. 13–204, 2014.
[7] N. Good and P. Mancarella, "Flexibility in multi-energy communities with electrical and thermal storage: A stochastic, robust approach for multi-service demand response," *IEEE Trans. Smart Grid*, vol. 10, no. 1, pp. 503–513, Jan. 2019.
[8] H. Lund, P. A. Østergaard, D. Connolly, and B. V. Mathiesen, "Smart energy and smart energy systems," *Energy*, vol. 137, pp. 556–565, 2017.
[9] M. Capone and E. Guelpa, "Implementing optimal operation of multi-energy districts with thermal demand response," *Designs*, vol. 7, no. 1, 2023, Art. no. 11.
[10] M. Capone, E. Guelpa, and V. Verda, "Optimal installation of heat pumps in large district heating networks," *Energies*, vol. 16, no. 3, 2023, Art. no. 1448.
[11] P. Mancarella, "MES (Multi-energy systems): An overview of concepts and evaluation models," *Energy*, vol. 65, pp. 1–17, 2014.
[12] M. Abugabbara, S. Javed, H. Bagge, and D. Johansson, "Bibliographic analysis of the recent advancements in modeling and co-simulating the fifth-generation district heating and cooling systems," *Energy Buildings*, vol. 224, 2020, Art. no. 110260.
[13] T. Godfrey, S. Mullen, D. W. Griffith, N. Golmie, R. C. Dugan, and C. Rodine, "Modeling smart grid applications with co-simulation," in *Proc. 1st IEEE Int. Conf. Smart Grid Commun.*, 2010, pp. 291–296.
[14] P. Mihal, M. Schvarcbacher, B. Rossi, and T. Pitner, "Smart grids co-simulations: Survey & research directions," *Sustain. Comput.: Inform. Syst.*, vol. 35, 2022, Art. no. 100726.
[15] P. R. Mazzarino, A. Macii, L. Bottaccioli, and E. Patti, "A multi-agent framework for smart grid simulations: Strategies for power-to-heat flexibility management in residential context," *Sustain. Energy, Grids Netw.*, vol. 34, 2023, Art. no. 101072.
[16] P. Nageler et al., "Co-simulation workflow for the dynamic modelling and simulation of large-scale district energy systems," in *Proc. 16th Conf. Building Simul.*, 2019, pp. 3698–3705.
[17] E. Vesaoja, H. Nikula, S. Sierla, T. Karhela, P. G. Flikkema, and C.-W. Yang, "Hybrid modeling and co-simulation of district heating systems with distributed energy resources," in *Proc. Workshop Model. Simul. Cyber-Phys. Energy Syst.*, 2014, pp. 1–6.
[18] H. Nikula et al., "Co-simulation of a dynamic process simulator and an event-based control system: Case district heating system," in *Proc. IEEE Emerg. Technol. Factory Automat.*, 2014, pp. 1–7.
[19] M. Capone, E. Guelpa, G. Mancò, and V. Verda, "Integration of storage and thermal demand response to unlock flexibility in district multi-energy systems," *Energy*, vol. 237, 2021, Art. no. 121601.
[20] P. R. Mazzarino, M. Capone, E. Guelpa, V. Verda, L. Bottaccioli, and E. Patti, "A multi-agent framework to evaluate energy flexibility in district heating networks," in *Proc. IEEE Int. Conf. Environ. Elect. Eng. IEEE Ind. Commercial Power Syst. Europe*, 2022, pp. 1–6.
[21] M. Capone, E. Guelpa, and V. Verda, "Potential for supply temperature reduction of existing district heating substations," *Energy*, vol. 285, 2023, Art. no. 128597.
[22] D. Müller, M. Lauster, A. Constantin, M. Fuchs, and P. Remmen, "AixLib-an open-source modelica library within the IEA-EBC annex 60 framework," in *Proc. BauSim Conf.*, 2016, pp. 3–9.
[23] V. Verda, M. Capone, and E. Guelpa, "Optimal operation of district heating networks through demand response," *Int. J. Thermodynamics*, vol. 22, no. 1, pp. 35–43, 2019.
[24] C. Steinbrink et al., "CPES testing with mosaik: Co-simulation planning, execution and analysis," *Appl. Sci.*, vol. 9, no. 5, 2019, Art. no. 923. [Online]. Available: https://www.mdpi.com/2076--3417/9/5/923
[25] F. Lu, Z. Yu, Y. Zou, and X. Yang, "Cooling system energy flexibility of a nearly zero-energy office building using building thermal mass: Potential evaluation and parametric analysis," *Energy Build.*, vol. 236, 2021, Art. no. 110763.