Energy & Buildings 292 (2023) 113146

Contents lists available at ScienceDirect

Energy & Buildings

journal homepage: www.elsevier.com/locate/enbuild

An ontology-based innovative energy modeling framework for scalable and adaptable building digital twins

Jakob Bjørnskov *, Muhyiddine Jradi

SDU Center for Energy Informatics, Mærsk Mc-Kinney Møller Instituttet, University of Southern Denmark, Campusvej 55, Odense M, 5230, Denmark

ARTICLE INFO

Keywords:
Digital twin
Data-driven
Building energy model
Building simulation
Ontology
SAREF

ABSTRACT

Digitalization of buildings and the use of IoT sensing and metering devices are steadily increasing, offering new opportunities for more autonomous, efficient, and flexible buildings. As part of this transformation and inspired by the added value demonstrated in other domains, the concept of a building digital twin that can monitor, simulate, manage, and optimize building operation has received increased interest. To aid such digital twin implementations, accurate and adaptable simulation models are required, which can effectively integrate and utilize the available data. However, traditional building modeling and digital practices, such as Building Information Modeling and white-box modeling tools, are not easily compatible with these requirements. This work presents an innovative and flexible energy modeling framework based on the SAREF ontology. With a basis in the SAREF4BLDG extension for buildings and the defined classes, different models are presented for a selection of typical systems and devices such as spaces, space heaters, dampers, coils, etc. Using the generic semantics and relations of the SAREF4SYST extension, a method for linking and simulating component models is then presented. A proof-of-concept of the modeling framework is provided, showing its application and feasibility to provide a dynamic simulation of the different systems and devices included in a demonstration case. Finally, a future line of work is identified considering the implementation of the modeling framework in an actual building case study, including integration with actual sensing equipment to demonstrate different digital twin services such as performance monitoring, strategy planning, and operational optimization.

1. Introduction

Driven by ambitious environmental goals and the promising potential for improved efficiency and automation by employing the Internet of Things (IoT) and Artificial Intelligence (AI) technologies, the building sector is currently undergoing fast-paced digitalization with increasing adoption of large sensing and metering networks. While the utilization of such data offers many opportunities, it also presents great challenges to status quo data management, building operation, and building energy modeling. As a consequence, the demand is steadily growing for more flexible and scalable data and energy models that integrate seamlessly.

As part of the digitalization of buildings, Building Information Models (BIM) have for the past two decades played a major role, unlocking significant cost savings and improving efficiency by easing communication and information exchange between contractors in the design and construction phases of buildings [1]. However, despite its demonstrated success, the BIM technology is now facing significant challenges as attempts are now made to extend it beyond its intended scope to fulfill the demands of IoT-integrated smart buildings. Boje et al. [2], argued that BIM is not suitable for IoT integration due to the static legacy formats and standards used to represent data. Instead, they highlighted the emerging concept of Digital Twins (DT) in buildings as a promising solution for solving these interoperability issues.

The DT concept was first used by the National Aeronautics and Space Administration (NASA), which defined it as a *"comprehensive multi-physical, multi-scale, probabilistic simulation system for vehicles or systems"* [3]. Following, the DT concept has been adapted in various other scientific and engineering fields such as product manufacturing, medical sciences, and smart cities [4]. Although DTs as a concept is starting to emerge in the building sector, the concept is still in its infancy with no clear consensus on a common definition and which services it should provide.

In this work, we adopt the definition provided by Boje et al. [2] and Grieves [5]. That is, a DT refers to a concept with three main constituents; a physical system, a virtual system, and a flow of data linking

* Corresponding author.
E-mail address: jabj@mmmi.sdu.dk (J. Bjørnskov).

https://doi.org/10.1016/j.enbuild.2023.113146
Received 22 January 2023; Received in revised form 1 May 2023; Accepted 3 May 2023
Available online 9 May 2023

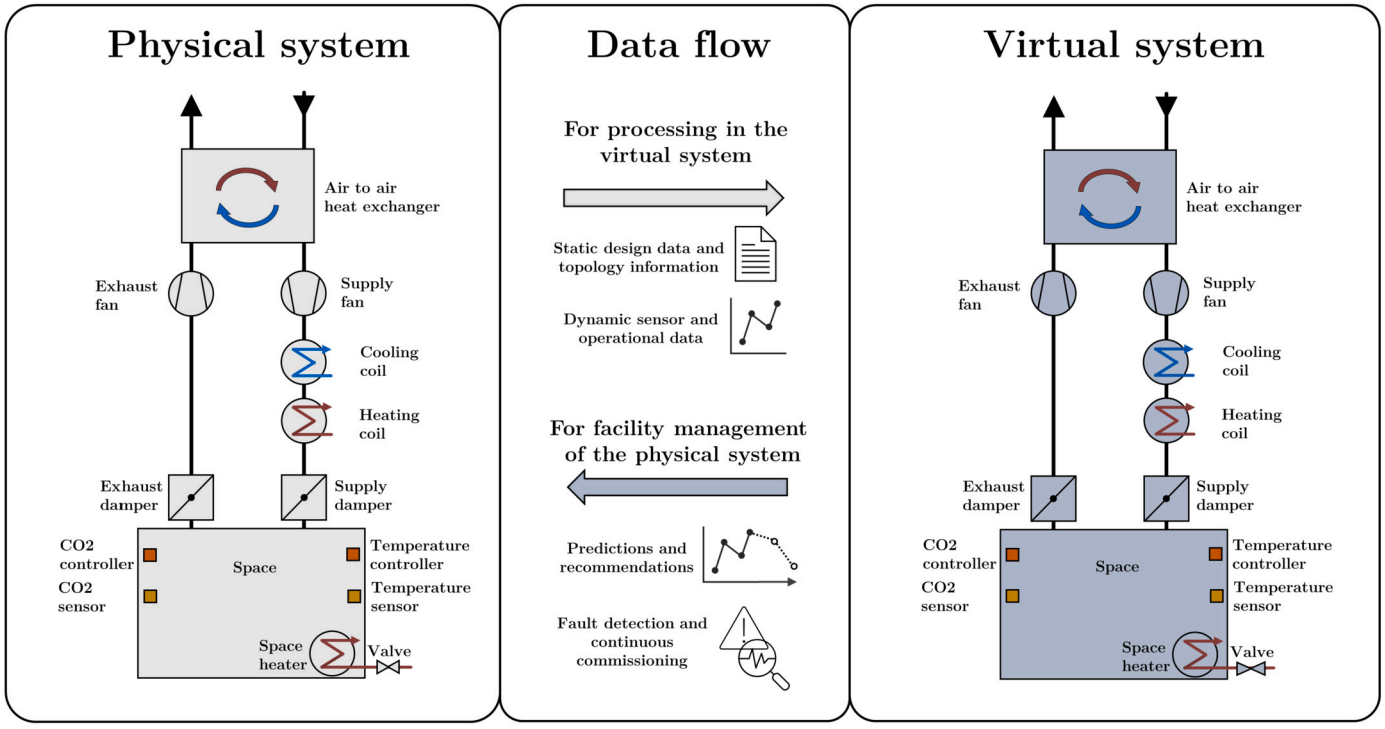

Fig. 1. The DT concept illustrated with typical systems and components present in buildings with their virtual counterpart and the bidirectional flow of data.

these two systems. This is conceptualized in the context of a generic building energy system as shown in Fig. 1. Here, the physical system is the actual asset to be managed, i.e. a building including all systems, devices, spaces, occupants, etc. The virtual system employs relevant simulation models to emulate the behavior of the physical system as closely as possible. The physical and virtual systems are linked through a bidirectional data flow, where the virtual system receives raw data in the form of static design and topology information as well as dynamic sensor and meter data. The virtual system then processes this information to adapt, monitor, simulate, and optimize based on the employed simulation models. This processed information is then communicated back to the physical system for facility management. As a basis for the work presented in this paper and the envisioned DT, three fundamental services, targeting the operational phase of the building life cycle, have been defined as follows:

Service 1 Efficient collection and management of data generated by sensors, meters, and IoT devices through a user-friendly open-standard context information model

Service 2 Automated continuous commissioning and performance monitoring in real-time to detect faults and malfunctions, ensuring a smarter and more cost-effective facility management

Service 3 Operational strategy planning support to enable more informed decision-making by running different control and management scenarios in a zero-risk virtual environment

In recent years, successful implementations of fault detection and performance monitoring, as listed in **Service 2**, have been reported numerous times with different approaches. Typically, fault detection methods are divided into two categories. The first category is model-based methods, which rely on comparisons between the simulated behavior of a system with actual measurements collected from the physical unit. The second category is model-free methods, which typically utilize only operational data collected from a system to find patterns that distinguish faulty from normal operation. Although model-free methods require no models to implement, their effectiveness is also often very limited in comparison to model-based methods when applied to large-scale distributed systems such as Heating Ventilation Air-Conditioning (HVAC) systems [6]. Therefore, most efforts in fault detection and continuous commissioning of buildings have considered model-based methods, often with the use of white-box simulation tools [7–10]. In addition, an accurate dynamic energy model enables simulation and optimization of future operational scenarios, as required by **Service 3**, aiding decision-makers with operating the building on a day-to-day basis, taking into account weather forecasts, expected occupancy, and price signals. This potentially increases indoor comfort, reduces operational costs, and improves the overall energy flexibility of the building.

The added value of the outlined services is well-established and has been demonstrated numerous times in case-studies [11–13]. However, using traditional energy modeling tools and being challenged by various interoperability issues, such implementations usually result in highly specialized solutions tailored to each specific building. This building-by-building approach, relies on manual workflows, is costly, hard to scale, and acts as a major barrier to broadly implementing such systems.

Therefore, inspired by the added value provided in different domains, there is a need to promote and develop DT technology for the building sector to harness the benefits provided by advanced sensing and metering devices and provide the outlined services at scale. As a key milestone in such promotion, an automated and adaptable framework for energy modeling of buildings is needed to effectively integrate such models into a building DT.

In this work, an innovative framework is proposed and presented for automated and adaptable energy model development to provide the simulation models required by building DTs. The framework builds upon the Smart Applications REFerence (SAREF) ontology [14] to ensure interoperability by using existing classes, concepts, and relations. A selection of SAREF classes is extended with adaptable data-driven gray-box and black-box models, to describe their dynamic behavior. Furthermore, a generic framework based on the SAREF4SYST extension is presented to describe how these component models interact based on simple topology information. To provide a proof-of-concept, a demonstration case is considered to illustrate the framework implementation considering the simple single-zone system shown in Fig. 1.

The presented framework in this work is capable to serve as a basis for the development and implementation of scalable DTs for building applications, allowing the delivery of major services to enhance the energy efficiency and intelligence quotient of buildings. This includes performance monitoring, data management, performance optimization, and strategy planning.

As established, one of the core elements of DTs is accurate simulation models. Therefore, in the following section, an overview and discussion of different approaches for building energy modeling are provided.

2. Building energy modeling

Over the years, considerable effort has been invested in developing large-scale energy simulation tools, e.g. EnergyPlus, DOE-2, and TRNSYS [15–17]. Models developed in such tools fall under the category of white-box models, which are based entirely on first-principles building physics, typically in the form of mass and energy balance equations, which often lead to large systems of Ordinary Differential Equations (ODE). However, being the case for the majority of large-scale white-box models, they require extensive information on the building and a substantial amount of resources and time to develop. Taking EnergyPlus as an example, a typical workflow consists of three time-consuming phases before an accurate energy model is obtained. First, the building geometry is defined with detailed geometry data from floor plans, cross-sections, 3D models, etc. Second, HVAC systems design data must be specified along with material and thermal properties of the envelope and occupancy, lighting, and equipment schedules of the spaces. Finally, to achieve acceptable prediction accuracy, the model must be calibrated using actual consumption data from the building, a process that often relies on a manual trial-and-error approach that requires domain-specific expertise from the modeler [18]. In addition to the high amount of resources and time that must be dedicated to the manufacturing of these high-resolution white-box models, high computational costs and execution time is also a well-known challenge, which makes them unsuitable for operational optimization purposes [19,20].

Therefore, to reduce the amount of manual work of white-box model development, considerable efforts have been put into the automatic translation of the design and geometrical data available in BIM to attain working Energy models for simulation tools (BIM2BEPS) such as EnergyPlus and Modelica. For instance, Ramaji et al. [21] proposed a BIM2BEPS transformation tool for conversion from the BIM-format International Foundation Classes (IFC) into an OpenStudio model.

Andriamamonjy et al. [22] proposed an automated BIM2BEPS workflow between IFC and Modelica. The workflow applies an intermediate representation called Model View Definition (MVD), on which different checks are performed to ensure a given IFC file includes all required information. If the IFC passes this test, it can be directly translated into a working Modelica model. While the workflow is successfully applied to a test case study, the authors also highlighted potential barriers of adoption. Namely, that it relies greatly on information that is typically not available in status quo IFC files.

Despite the significant efforts in BIM2BEPS research, many of the existing interoperability issues such as interdependence on extensive toolchains, inconsistent current practices by BIM practitioners, and loss of information still persist [23,24].

As a result of these shortcomings, data-driven modeling methods have recently gained increasing popularity for building modeling and simulation, due to their lower computational demands and a high potential for automation and integration in smart buildings where sensor networks and metered HVAC components collect large amounts of operational data.

Data-driven modeling methods are typically divided into black-box and gray-box methods. Black-box models are purely data-driven, meaning that no domain-specific model structure is assumed. Given a set of data, specific algorithms are thus applied to find appropriate functional relationships that map system inputs to system outputs. Commonly used black-box models are Artificial Neural Networks (ANN) [25,26], Support Vector Machines (SVM) [27], Decision Trees [28], and state-space models [29]. Due to the absence of domain-specific assumptions, these types of models have been successfully applied across many academic disciplines as well as in industry [30,31].

Typically, black-box models are considered lighter than white-box models in terms of computational complexity and simulation speed. For existing buildings with available historical data, black-box models have been used for aggregated heating and electricity load forecasting [32–34]. These types of models are usually referred to as *monolithic models*, i.e. whole-building models that do not incorporate any knowledge of the modeled building, besides historical weather and operational data such as heating or electricity consumption. While this offers convenient and fast implementation, there are also certain drawbacks to consider. If the building operation or design is changed, e.g. if system setpoints are changed or if the building is retrofitted, new data have to be collected and the model has to be re-trained. Furthermore, if the data available is scarce or of low quality, the performance of black-box models might be affected. A promising solution to this issue is the use of *transfer learning*, a technique where knowledge is transferred from one domain to another [35]. Thereby, the knowledge contained in models that are trained on large amounts of data can be reused for training other models, given that the domains have a certain amount of overlap. This concept has been applied for both indoor temperature prediction [36] and energy prediction [37].

One of the often highlighted drawbacks of black-box models is the lack of interpretability, i.e. the model structure and the obtained parameters have no direct physical meaning and the underlying cause of model behavior is thus unknown. This is opposed to gray-box models, the second category of data-driven models. As the name suggests, these types of models are a mix of white-box and black-box models where the model structure is based on physical principles, while the unknown parameters of this model structure are estimated through parameter identification techniques. The physics-derived model structure enables interpretation and validation of the obtained parameters from parameter estimation in a physical context, e.g. as demonstrated by Macarulla et al. [38]. An example of the use of gray-box modeling in buildings is the thermal Resistor Capacity (RC) model, a thermal analog to electrical circuits used to model the thermal dynamics of buildings [39,40]. However, gray-box modeling is a general concept that can also be applied to individual building components such as heat pumps, recovery units, thermal storage tanks, cooling coils, etc., as demonstrated by Afram et al. [41].

In general, the various recent studies on data-driven modeling of building systems and components using black-box and gray-box approaches report clear benefits regarding automation potential and flexibility as well as high prediction accuracy. Considering the discussed challenges of BIM2BEPS automation, these are key properties for applications in a dynamic DT-environment where close integration between data collected from IoT networks and simulation models is fundamental for delivering the outlined DT services.

3. Ontologies and metadata schemes for improved interoperability and building energy modeling

Buildings have a high degree of heterogeneity and complexity, including a wide variety of systems and devices from different contractors and providers. This makes it difficult to develop applications and models that can be scaled and reused across different buildings and use-cases. This has caused the development of ontologies, which attempts to standardize and structure metadata and communication between devices to avoid the status-quo many-to-many translation between the multitude of protocols used.

Pritoni et al. [42] provided a review of 40 different ontology and metadata schemas for building design, energy modeling, and building

operation applications. From these 40 different ontologies, five of the most popular ontologies were selected for a deeper comparison based on their applicability in three use cases; Energy auditing, automated fault detection and diagnosis, and optimal control of HVAC systems. The selected ontologies were SAREF, Semantic Sensor Network/Sensor, Observation, Sample, and Actuator (SSN/SOSA), Building Topology Ontology (BOT), Brick, and RealEstateCore (REC).

Generally, it was highlighted that all five ontologies had varying degrees of missing concepts, which impacted their utility in the three use cases, i.e. choosing the right ontology depends on its intended application. For instance, BOT and SSN/SOSA missed central concepts for describing sensed properties, actuator idioms, and units of measurement. BOT, Brick, and REC missed equipment properties to describe e.g. nominal capacities, flowrates, efficiencies, etc. All ontologies except Brick missed schedule concepts for describing control strategies. Generally, Pritoni et al. conclude that none of the five ontologies accounts for all required concepts and that the ontologies must be customized and extended for certain applications.

In this work, SAREF is chosen as a backbone for the energy modeling framework. SAREF was developed in 2015 by Daniele et al. [43] in close collaboration with industry to define a unified reference for recurring concepts and relations in the IoT domain. The core SAREF ontology revolves around the concept of devices, e.g. a lighting switch, sensor, or meter. The ontology has extensions spanning across multiple domains, e.g. SAREF4BLDG [44], SAREF4CITY [45], SAREF4SYST [46], SAREF4ENER [47], etc. Here, SAREF4BLDG is an extension dedicated to the building domain and is based on the ISO-published Industry Foundation Classes (IFC) standard [48]. The extension has 72 classes and 179 object properties that represent typical appliances and devices in buildings. Hence, this ontology allows an almost direct mapping between IFC and SAREF for most of these objects within the building domain. This is a great advantage compared with other relevant ontologies, which could potentially allow for significant reuse of existing well-structured information, contained in the frequently available IFC models for buildings. Another ontology that also has IFC mapping capabilities is IFCowl [49]. However, as discussed by Rasmussen et al. [50], IFCowl was designed to be backwards compatible with IFC and as a result inherits two main drawbacks; complexity and size. This makes the ontology both hard to understand and hard to use. In comparison, SAREF has a higher degree of modularity and its extension SAREF4BLDG is lighter in complexity and size. In addition, the multiple domains and broader scope of SAREF could allow for further expansion of the DT concept to consider larger systems such as districts or cities at a later stage.

As is the case with the components in IFC, the component classes specified in SAREF4BLDG are data containers with no formal or mathematical description of their dynamic behavior or interaction with other components or systems. For instance, a `s4bldg:SpaceHeater` object holds the property $\texttt{s4bldg:outputCapacity}^{\texttt{op}}$, but it has no attached model which specifies how this property should be used to simulate actual heat consumption.

Therefore, to build upon the SAREF4BLDG extension and ensure high flexibility, a modular and component-based modeling approach is proposed. Using the component definitions and class hierarchies from SAREF4BLDG, the energy model is thus split into a matching set of component models that can be automatically assembled and adapted, based on the specific use case. Each component should be able to learn from the collected data through various parameter estimation techniques and strive to improve prediction accuracy as the collected data accumulates. This approach will make it possible to continuously monitor and compare the performance of actual components in real-time with their digital counterparts in the digital twin environment, as required by **Service 2**. Furthermore, a highly modular and flexible model allows for the interchange of component models during the entire building lifecycle, which is necessary to maintain the Digital Twin as a close replica of the actual building, as the installed systems or building use might change over time. Therefore, the component models should be accurate enough to meet the defined services to a satisfactory level, but simple enough to allow for robust and automated model calibration.

With a basis in the components depicted in Fig. 1, a description of the components and their properties from the SAREF4BLDG ontology is given in the Appendix Table B.3. The superscripts `op` and `dp` stand for object property and data property, respectively. An object property points to another object, while a data property holds data in the form of standard types such as integer, string, floating-point, etc. A complete list of available components and a description of the listed properties can be found in the SAREF4BLDG documentation [44].

4. Component modeling

When modeling buildings, it is common practice to take advantage of the major difference in physical time constants that are associated with the HVAC system and the building envelope and spaces. The HVAC system is often governed by fast-moving dynamics that reach equilibrium within seconds or minutes, while the large thermal inertia of the building envelope and spaces has very slow-moving dynamics. Therefore, it can often be justified to model the HVAC systems as quasi-steady-state systems, which decreases model complexity and lowers the computational burden considerably [51,52]. For these reasons, building simulation tools such as EnergyPlus is also heavily based on this modeling approach [53]. Therefore, with a basis in the presented components in Table B.3, potential candidate models have been found in the literature for implementation and detailed descriptions are included in Appendix A. The key selection criteria for the models have been simplicity and applicability for data-driven calibration and parameter estimation.

To provide an overview of all the identified models, the inputs, outputs, parameters, and constants have been summarized for each model in Table 1. The parameters for the models in Table 1 can either be given through design specifications or calculated through parameter estimation techniques if time-series data is available for the inputs and outputs. In the case where input-output data can be continuously provided, parameter estimation can be performed on a regular interval to attain a continuously improving performance on the different gray- and black-box component models. This method is demonstrated for indoor temperature forecasting by Ruano et al. [54]. Additionally, each physical component can be monitored closely, by comparing its operation with its digital replica to provide fault detection and continuous commissioning capabilities on a component level. If neither design specifications nor operational input-output data is available, default values must be assumed.

Based on the listed inputs and outputs in Table 1, and the model descriptions in the previous sections, certain expected relationships between components arise. For instance, the space heater model in Table 1 expects the massflow $\dot{m}_w$, and the room temperature T_z as input, which it can acquire from the valve model and the building space model, respectively. The valve model expects the valve position u_v as input, which it can acquire from the controller component etc. In the following section, these types of relations are formalized using simple but broadly applicable concepts from the SAREF4SYST extension.

5. Modeling framework

In the previous section, common building components were presented, along with generic and scalable component energy models. However, to express how the defined components interact, a domain-independent SAREF-compliant methodology for linking the inputs and outputs of component models is defined. The concept of having different self-contained component models defined by an input-output relationship and an overall framework that links these models share many similarities to the co-simulation framework defined in the Functional Mockup Interface (FMI) standard [55]. Here, models are exported from simulation tools as Functional Mockup Units (FMU) and are used as

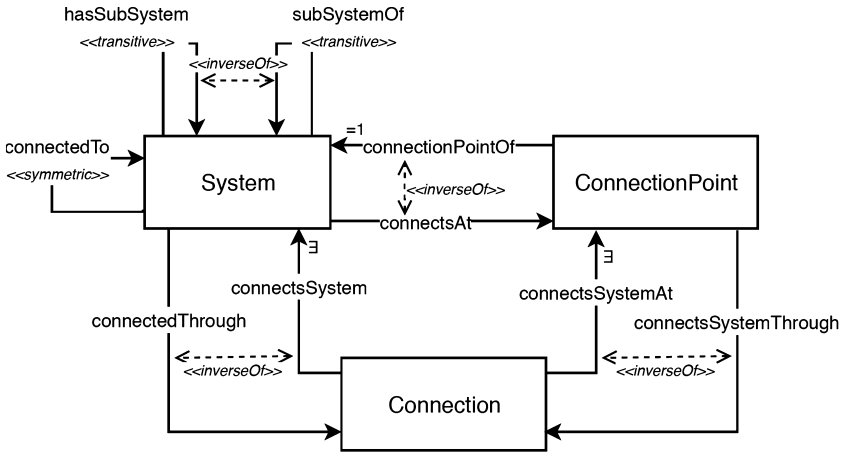

Fig. 2. Overview of the SAREF4SYST ontology extension with available classes and properties [46].

Table 1
Overview of inputs, outputs, parameters, and constants of potential mathematical models for the highlighted components.

Component	Inputs	Outputs	Parameters	Constants
Valve	$u_v \in [0,1]$	$\dot{m}_w \in \mathbb{R}^+$	$N_v \in [0,1]$ $\dot{m}_{w,max} \in \mathbb{R}^+$	
Space heater	$\dot{m}_w \in \mathbb{R}^+$ $T_{w,in} \in \mathbb{R}$ $T_z \in \mathbb{R}$	$\dot{Q}_h \in \mathbb{R}^+$	$UA \in \mathbb{R}^+$ $C_r \in \mathbb{R}^+$	$c_{p,w}$ Δt
Heating coil	$\dot{m}_a \in \mathbb{R}^+$ $T_{a,set} \in \mathbb{R}$ $T_{a,in} \in \mathbb{R}$	$\dot{Q}_{hc} \in \mathbb{R}^+$		$c_{p,a}$
Cooling coil	$\dot{m}_a \in \mathbb{R}^+$ $T_{a,set} \in \mathbb{R}$ $T_{a,in} \in \mathbb{R}$	$\dot{Q}_{cc} \in \mathbb{R}^+$		$c_{p,a}$
Air to air heat recovery	$\dot{m}_{a,sup} \in \mathbb{R}^+$ $\dot{m}_{a,exh} \in \mathbb{R}^+$ $T_{a,sup,in} \in \mathbb{R}$ $T_{a,exh,in} \in \mathbb{R}$ $T_{a,set} \in \mathbb{R}$	$T_{a,sup,out} \in \mathbb{R}$	$\epsilon_{75\%,h} \in [0,1]$ $\epsilon_{75\%,c} \in [0,1]$ $\epsilon_{75\%,h} \in [0,1]$ $\epsilon_{100\%,c} \in [0,1]$ $\dot{m}_{a,max}$	$c_{p,a}$
Fan	$\dot{m}_a \in \mathbb{R}^+$	$\dot{W}_{fan} \in \mathbb{R}^+$	$c_1 \in \mathbb{R}$ $c_2 \in \mathbb{R}$ $c_3 \in \mathbb{R}$ $c_4 \in \mathbb{R}$ $\dot{m}_{a,max}$ ΔP_{max} $\eta_{tot} \in [0,1]$	ρ_a
Damper	$u_d \in [0,1]$	$\dot{m}_a \in \mathbb{R}^+$	$a \in \mathbb{R}$ $b \in \mathbb{R}$ $c \in \mathbb{R}$	
Controller	$y_{set} \in \mathbb{R}$ $y_{meas} \in \mathbb{R}$	$u \in [0,1]$	$K_p \in \mathbb{R}$ $K_i \in \mathbb{R}$ $K_d \in \mathbb{R}$	
Building Space	$N_{occ} \in \mathbb{N}$ $\dot{m}_{a,sup} \in \mathbb{R}^+$ $\dot{m}_{a,exh} \in \mathbb{R}^+$ $T_o \in \mathbb{R}$ $\Phi_s \in \mathbb{R}^+$ $u_v \in [0,1]$ $u_d \in [0,1]$ $u_s \in [0,1]$	$C_z \in \mathbb{R}$ $T_z \in \mathbb{R}$	$K_{occ} \in \mathbb{R}^+$ $m_z \in \mathbb{R}^+$ [-]	Δt

so-called slaves. The slaves are allowed to communicate at specific time intervals called communication points, managed by a master algorithm. Between communication points, the slaves are simulated independently. The modeling framework presented in this section shares many of the same principles as co-simulation and is thus well-aligned with the use of FMUs for modeling individual components.

5.1. Linking component models using SAREF4SYST

SAREF has a dedicated extension called SAREF4SYST, which aims at providing a generic framework for representing the topology of systems and how they interact [46]. Through SAREF4SYST, the system topology can be expressed through three generic classes, `s4syst:System`, `s4syst:Connection`, and `s4syst:ConnectionPoint` along with nine applicable properties as shown in Fig. 2.

As shown in Fig. 2, a `s4syst:System` object can be a subsystem of other `s4syst:System` objects specified through the property `s4syst:subSystemOf` or have subsystems itself specified through `s4syst:hasSubSystem`. Various forms of flows between systems, e.g. information or physical quantities can be expressed through the `s4syst:Connection` and `s4syst:ConnectionPoint` classes. In this work, the `s4syst:Connection` class is used to represent system outputs and the `s4syst:ConnectionPoint` class to represent system inputs. Hence, all inputs and outputs in Table 1 can be represented by `s4syst:ConnectionPoint` and `s4syst:Connection` objects respectively, while each component can be represented by `s4syst:System` objects. This concept is shown in Fig. 3, where three SAREF4BLDG components are shown. As recommended in the SAREF4BLDG documentation [56], each component is extended with its identified model as a specialized subclass from section 4. Furthermore, these components all inherit from the `s4syst:System` superclass, enabling them to interact and communicate through the SAREF4SYST ontology. It should be noted that the full SAREF4BLDG class hierarchy is not included for clarity.

In Fig. 3, simple information flows between the three components are shown. The `s4bldg:BuildingSpace` component points to a `s4syst:Connection` object through the `s4syst:connectedThrough` property. This `s4syst:Connection` object represents the indoor temperature output T_z of the `s4bldg:BuildingSpace` model extension. Through the `s4syst:connectsSystemAt` property, it further points to a `s4syst:ConnectionPoint` object, which represents the input T_z of the `s4bldg:SpaceHeater` model extension. This relation is expressed through the `s4syst:ConnectionPoint` property of the `s4syst:ConnectionPoint` object. The same concepts are then applied to relate the `s4bldg:Valve` water massflow output $\dot{m}_w$ to the `s4bldg:SpaceHeater` input $\dot{m}_w$. Altogether, this example shows how information is translated from model output to model input. Correspondingly, the inverse path from model input to model output is expressed through the `s4syst:connectsAt`, `s4syst:connectsSystemThrough`, and `s4syst:connectsSystem` properties as shown in Fig. 2. This generic framework and the translation from output to input and vice versa at the `s4syst:Connection` and `s4syst:ConnectionPoint` objects provide the flexibility to exchange information between any type of models independently of model formulation and

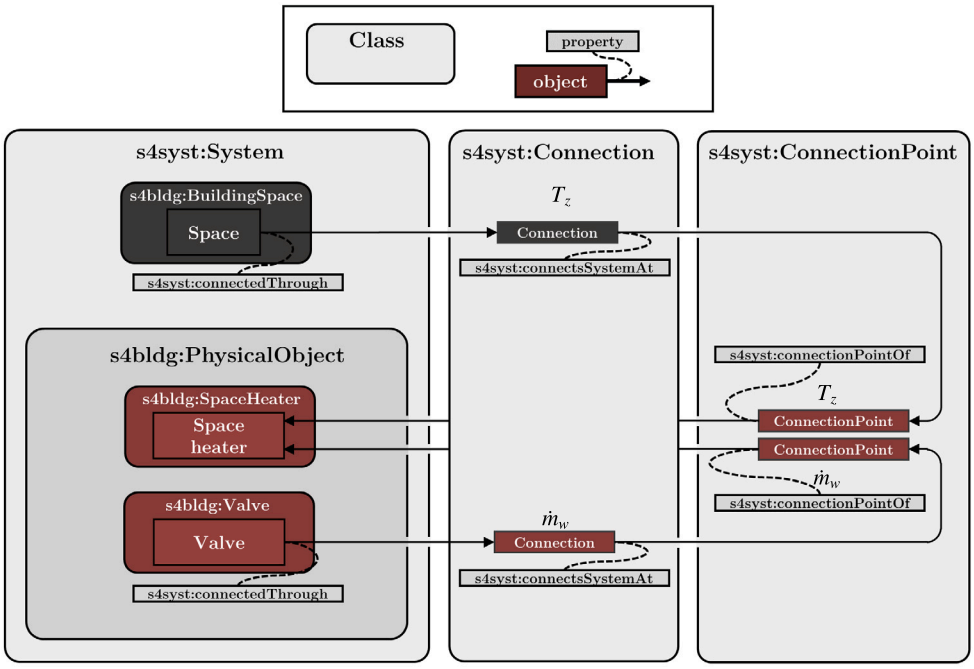

Fig. 3. Example of data structure showing how SAREF4SYST is used to link models contained in the SAREF4BLDG components through the `s4syst:System`, `s4syst:Connection`, and `s4syst:ConnectionPoint` classes.

input-output terminology. Hence, the presented framework is not specific to the models shown in Table 1, which could be modified or extended to cover other components such as `s4bldg:Chiller` or `s4bldg:Pump`, where models from external simulation tools and libraries could be considered, e.g. as FMUs.

5.2. Model execution order

The topology in Fig. 3 defines the required information exchange between components at a given timestep. This produces a scheduling problem, where it must be ensured that all required inputs are available when a model is executed during a simulation time step. For instance, in Fig. 3, both the `s4bldg:BuildingSpace` and the `s4bldg:Valve` model must be executed before the `s4bldg:SpaceHeater` model can be executed. Such task scheduling problems are well-known in computer science and are formally known as topological sorting [57]. Given a directed graph $G = (V, E)$, with V being the set of nodes and E being the set of edges, topological sorting is the ordering of nodes such that for every edge $(u, v) \in E$ the node u comes before the node v. This is only possible if the graph has no cycles, i.e. if it is a Directed Acyclic Graph (DAG), in which case it is guaranteed to have at least one solution [57,58]. Hence, if the `s4syst:System` instances are interpreted as nodes, and the `s4syst:Connection`, and `s4syst:ConnectionPoint` pairs are interpreted as directed edges, the correct execution order can be found using topological sorting, given that the graph is acyclic.

5.3. Cycle removal procedure

The no-cycles condition of topological sorting is essential to consider when applying the presented methodology to model actual buildings as feedback control loops are an integral part of all building automation systems. When such control loops are modeled, cycles are introduced. An example of a cycle produced by a controller is shown in Fig. 4. Here, a `s4bldg:BuildingSpace` model predicts CO_2-concentration C_z, which is used as input y in a `saref:Sensor` model. The output y_{meas} is then used as input in a `s4bldg:Controller` model. An appropriate input signal u is then computed, which is sent to a `s4bldg:Damper` model to calculate the resulting airflow m_a. Finally, this value is sent back to the `s4bldg:BuildingSpace` model for recalculation of indoor CO_2-concentration C_z at the next time step, completing the cycle. To systematically deal with such cycles, an algorithm is proposed and is presented in Algorithm 1. Given the set of all `s4syst:System` objects H and an initially identical set H^*, the algorithm iterates through the set of all `s4bldg:Controller` components $U \in H^*$, identifies reachable components V_u from $u \in U$, and removes all edges from $v \in V_u$ to the controlled component `u.controlsProperty.isPropertyOf` if such edges exist. For instance, applying Algorithm 1 on the simple system shown in Fig. 4 the `Supply damper` would be identified as a reachable component from the `CO2 controller`. The `Supply damper` is connected with the controlled component `Space`, and this connection is thus removed.

5.4. Topological sorting procedure

After applying Algorithm 1, topological sorting can be applied on the modified set of components H^*. This is shown in Algorithm 2 where an adaptation of Kahn's topological sorting algorithm is presented, which matches the semantics of SAREF4SYST. The input of this algorithm is the set H^*, while the output is a sequence L with all the components in H in a topologically sorted order. The algorithm recursively adds all mapped components $f(s) \in H$ with $|s.\text{connectsAt}| = 0$ (i.e. the component s has no inputs) to a sequence L while removing all edges originating from s. In addition, a sequence P is created which holds integers starting from 0 that represent the execution priority of the components in L. Components in L with the same priority can be executed in parallel. For instance, applying Algorithm 2 on the simple system shown in Fig. 3 produces the sequence $L = (\texttt{Valve}, \texttt{Space}, \texttt{Space heater})$ and the priority sequence $P = (0, 0, 1)$. This implies that the `Valve` and `Space` models can be executed in parallel and that they must be executed before the `Space heater` model.

5.5. Simulation procedure

When the execution sequence L is found using Algorithm 2, the system can be simulated by simply traversing through this sequence

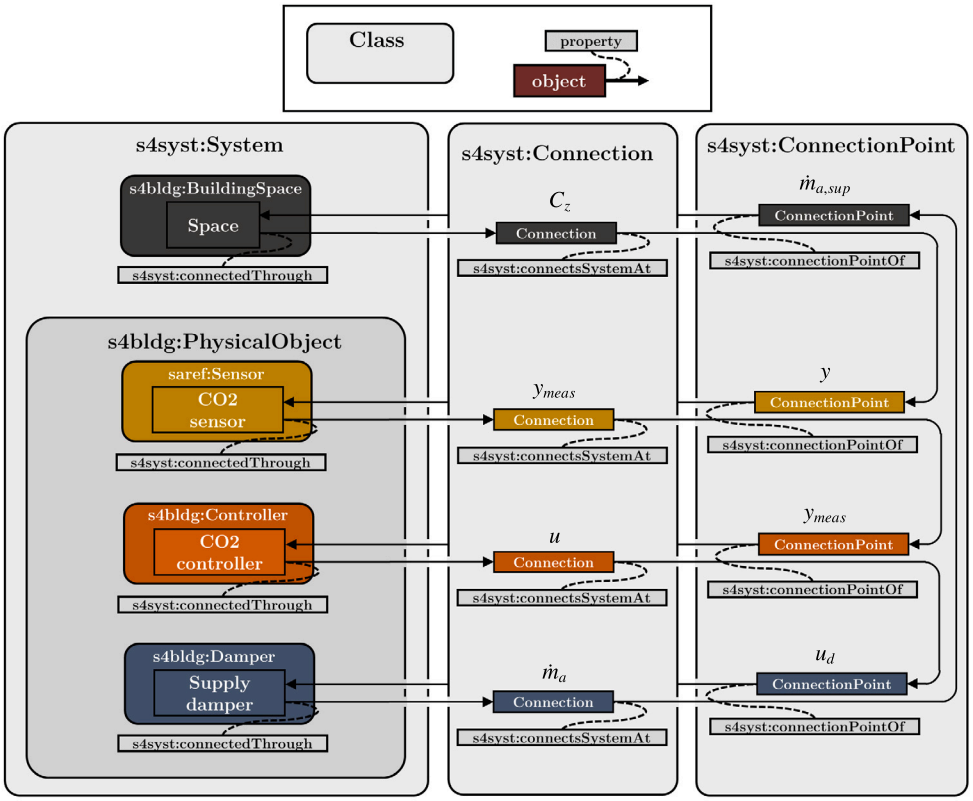

Fig. 4. Example of a cycle produced by a controller, which actuates the opening position of a supply airflow damper of a space. Some model inputs are neglected for clarity (e.g. the setpoint for the controller).

Algorithm 1 Removes cycles created by feedback control loops. DFS(V, s) is a recursive depth-first search to find the set of reachable components V from the component s.

Let H be the set of all instances with superclass `s4syst:System`
Let H^* contain a copy of all the elements in H such that the elements in H^* can be modified to contain no cycles
Let U be the subset of all instances with the class `s4bldg:Controller` of H^*

function DFS(V, s)
  $V \leftarrow V \cup \{s\}$
  for all $c \in s$.connectedThrough **do**
    $cp \leftarrow c$.connectsSystemAt
    $s^* \leftarrow cp$.connectionPointOf
    if $s^* \notin V$ **then**
      $V \leftarrow$ DFS(V, s^*)
    end if
  end for
  return V
end function

for all $u \in U$ **do**
  $w_u \leftarrow u$.controlsProperty.isPropertyOf
  $V_u \leftarrow$ DFS($\emptyset$, u)
  for all $s \in V_u$ **do**
    for all $c \in s$.connectedThrough **do**
      $cp \leftarrow c$.connectsSystemAt
      $s^* \leftarrow cp$.connectionPointOf
      if $w_u = s^*$ **then**
        w_u.connectsAt $\leftarrow w_u$.connectsAt $\setminus \{cp\}$
        s.connectedThrough $\leftarrow s$.connectedThrough $\setminus \{c\}$
      end if
    end for
  end for
end for

Algorithm 2 A SAREF4SYST-specific adaptation of Kahn's topological sorting algorithm [59].

Let $f : H^* \rightarrow H$ be a function that maps from the modified components $h^* \in H^*$ to the original components $h \in H$
Let S be the subset of H^* with no inputs, i.e. $S = \{s \in H^* : |s.\text{connectsAt}| = 0\}$
Let $L \leftarrow ()$ be an empty sequence which will be populated with the components of H in a sorted order
Let $P \leftarrow ()$ be an empty sequence which will be populated with the priority of the components in the sequence L
Let $p \leftarrow 0$ be the priority counter
The notation $X^\frown(x)$ is used to append the element x to the sequence X

while $|S| > 0$ **do**
  $S^* \leftarrow \emptyset$
  for all $s \in S$ **do**
    $L \leftarrow L^\frown(f(s))$
    $P \leftarrow P^\frown(p)$
    for all $c \in s$.connectedThrough **do**
      for all $cp \in c$.connectsSystemAt **do**
        $s^* \leftarrow cp$.connectionPointOf
        s^*.connectsAt $\leftarrow s^*$.connectsAt $\setminus \{cp\}$
        if $|s^*.\text{connectsAt}| = 0$ **then**
          $S^* \leftarrow S^* \cup \{s^*\}$
        end if
      end for
    end for
  end for
  $p \leftarrow p + 1$
  $S \leftarrow S^*$
end while

of components, while gathering the needed inputs and executing the attached model. Executing this routine once completes one system timestep of length Δt and can be repeated indefinitely given the required inputs, such as weather data and schedule values, are available. This is formulated in Algorithm 3, where each component model is assumed to have an input and output dictionary which are indexed by the properties `inputName` and `outputName` of the associated `s4syst:ConnectionPoint` and `s4syst:Connection` objects, respectively. These property names are simply the mathematical notation

for the inputs and outputs shown in Table 1 translated into a representative name, e.g. “indoorTemperature” to represent T_z. In addition, each component model is assumed to have a `do_step()` method, implementing the governing equations defined in section 4 or executing external models or libraries such as exported FMU models.

Algorithm 3 An algorithm for simulating building devices and systems given the execution sequence L.

Let t_{start} and t_{end} be the start and end simulation time
Let Δt be the simulation time step
$t \leftarrow t_{start}$
while $t < t_{end}$ **do**
  for $i \leftarrow 1 \dots |L|$ **do**
    $s \leftarrow L_i$
    for all $cp \in s$.connectsAt **do**
      $c \leftarrow cp$.connectsSystemThrough
      $s^* \leftarrow c$.connectsSystem
      s.input[cp.inputName] $\leftarrow s^*$.output[c.outputName]
    end for
    s.output $\leftarrow s$.do_step(s.input)
  end for
  $t \leftarrow t + \Delta t$
end while

6. Demonstration case and implementation

To validate the proposed modeling framework, the single-zone system shown in Fig. 1 is considered as a demonstration case. The modeling framework is implemented in Python as a reusable library [60], following the semantic structure and guidelines provided by the SAREF ontology. However, to accurately model the system, basic topological information is required. To ensure that the model can easily be constructed and adapted without manually specifying all the logical connections between models as presented in section 5, a simple model-independent representation is instead used as a starting point. Based on this representation, the logical connections between models can then be inferred based on simple rules. For this purpose, the single-zone system has been formulated using the SAREF ontology semantics and concepts as shown in Fig. 5 only considering the actual topology of the physical system.

6.1. Available topology information

The system consists of one `s4bldg:BuildingSpace` instance, containing one `s4bldg:SpaceHeater`, one `s4bldg:ShadingDevice`, one `s4bldg:Valve`, two `s4bldg:Damper`, two `s4bldg:Controller`, and two `saref:Sensor` instances as represented by the `s4bldg:contains` property. The `s4bldg:BuildingSpace` instance is linked with one `saref:Temperature` instance representing indoor temperature and one `saref:CO2` instance representing CO2-concentration through `saref:hasProperty`, which is inherited through the `saref:FeatureOfInterest` super class. The `saref:CO2` subclass of `saref:Property` has been added as an extension to the core SAREF ontology similar to the work carried out by Weerdt et al. [61], as it is not included by default. The measured properties of the two `saref:Sensor` instances are then determined through the `saref:measuresProperty`, pointing either to the `saref:Temperature` instance or the `saref:CO2` instance. Similarly, fashion, the controlled properties of the `s4bldg:Controller` instances are determined through the `saref:controlsProperty`. To determine which devices are actuated by the controllers, each `s4bldg:Controller` instance is associated with a `saref:LevelControlFunction` instance (subclass of `saref:Function`) through the `saref:hasFunction` property, which further points to a `saref:SetLevelCommand` instance (subclass of `saref:Command`) through the `saref:hasCommand` property. Finally, the `saref:SetLevelCommand` instances each point to a `saref:MultiLevelState` instance (subclass of `saref:State`) through the `actsUpon` property. The specific subclasses used here indicate that the actuated `s4bldg:Damper` and `s4bldg:Valve` instances can be regulated with a level-setting, e.g. from 0-100%. Alternatively, for representing binary control of other objects such as windows or doors, the `saref:OpenCloseFunction`, `saref:OpenCommand`, `saref:CloseCommand`, `saref:OpenState`, and `saref:CloseState` could be used instead. For the shown system, the supply and exhaust dampers share the same state, resulting in balanced supply and exhaust airflows. However, for controlling supply and exhaust airflows separately, each damper instance should have its own `saref:MultiLevelState` instance each associated with two different `s4bldg:Controller` instances.

To describe the topology of the ventilation system components and their order of placement in the flow path, the `s4syst:connectedTo` property is used, which is inherited from the `s4syst:System` class. This property is symmetric, meaning that two connected components both point to each other with the `connectedTo` property (although only one path is shown in Fig. 5). Hence, the `s4bldg:AirToAirHeatRecovery` instance is connected to both the supply and exhaust `s4bldg:Fan` instances while the exhaust fan is further connected to an exhaust `Node` instance. Here, the `Node` class represents nodes in a flow network and keeps track of flow and temperature before one stream branches out or after multiple streams join together. In the demonstration case, there is only one space, resulting in the same flow properties before and after the `Node` instances. However, this addition is necessary in the case of multiple spaces and a more complex flow path topology. The exhaust `Node` instance is further connected to an exhaust `s4bldg:Damper` instance. The supply `s4bldg:Fan` is connected to the heating `s4bldg:Coil` instance, which is connected to a supply `Node` instance. Finally, this supply `Node` instance is connected to a supply `s4bldg:Damper` instance.

For Fig. 5, inverse properties also exist for most of the shown properties, although these are omitted for clarity. For example, `s4bldg:isContainedIn` and `saref:isMeasuredByDevice` are inverse properties of `s4bldg:contains` and `saref:measuresProperty`.

6.2. Obtaining a working energy model

With the SAREF representation in Fig. 5 of the considered single-zone system, the topological context of all devices and components is defined. However, this representation cannot be directly used with the presented algorithms in section 5 to simulate the system. Therefore, inputs and outputs of each model extension, as described in section 5, are first matched using the topological relationships shown in Fig. 5 and the `s4syst:Connection` and `s4syst:ConnectionPoint` classes as demonstrated in section 5.

For example, a connection between the `s4bldg:BuildingSpace` instance and the CO2 `saref:Sensor` instance is inferred through the properties `saref:measuresProperty` and `saref:isPropertyOf` of the `saref:Sensor` and `saref:CO2` instance, respectively. A connection between the `saref:Sensor` and `s4bldg:Controller` is established through the `saref:controlsProperty` and `saref:isMeasuredByDevice` properties.

The resulting connections are shown in Fig. 4. Completing this process for all `s4syst:System` instances produces the graph shown at the top of Fig. 6. Here, each pair of `s4syst:Connection` and `s4syst:ConnectionPoint` instances linking two models are represented as one edge labeled with the shorthand “C:” and “CP:” representing the `outputName` and `inputName` properties, respectively. As mentioned previously, these property names correspond to the mathematical notation of input and outputs in Table 1. In addition to the already discussed `Node` class, two other helper classes `OutdoorEnvironment`, and `Schedule` have also been introduced. `OutdoorEnvironment` represents relevant outdoor conditions such as outdoor temperature and solar irradiation, which affects different systems, in this

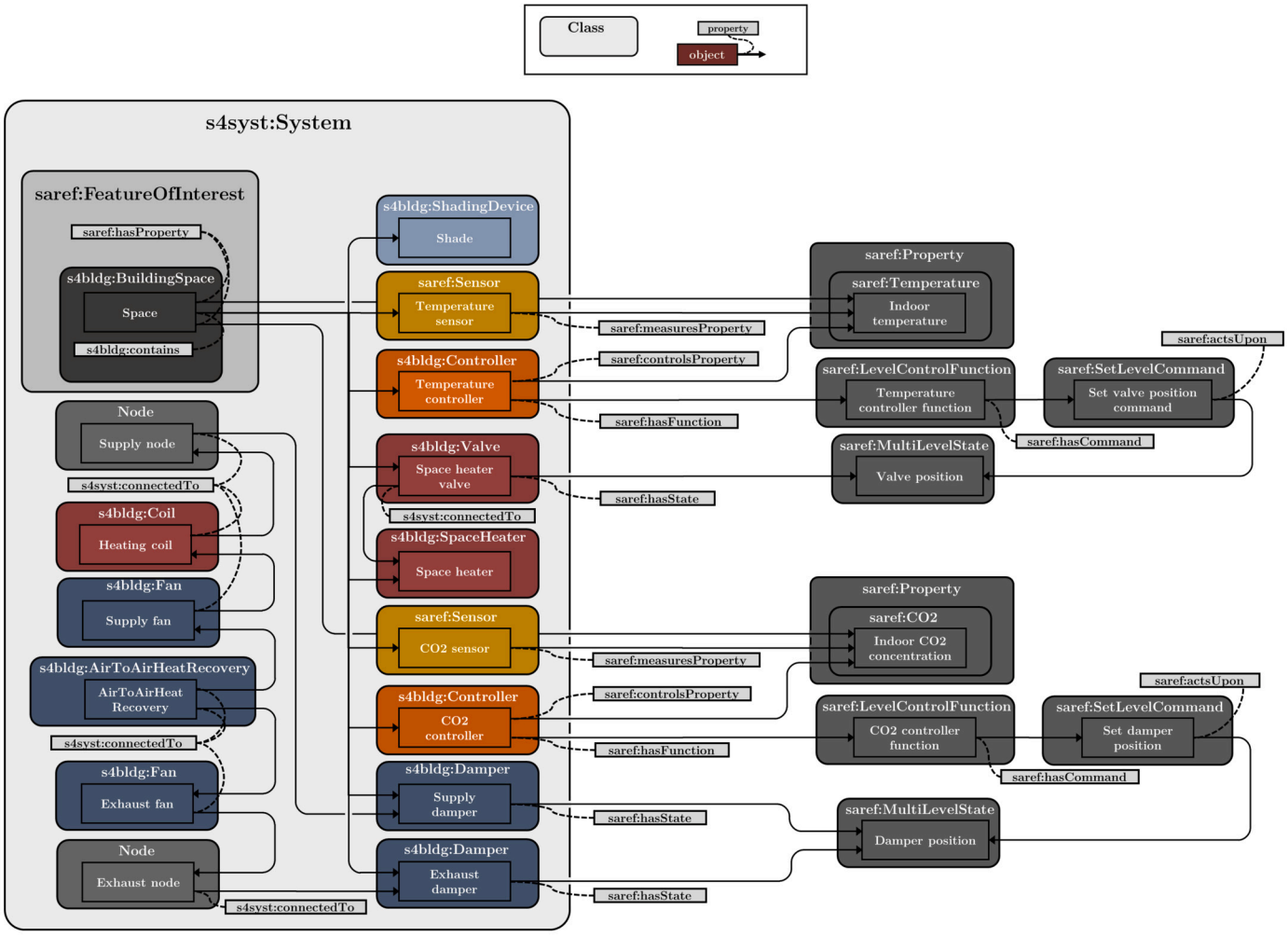

Fig. 5. SAREF representation of the demonstration case, including the required topology information for the involved components. The representation incorporates concepts from both SAREF core, SAREF4BLDG, and SAREF4SYST.

case, the `s4bldg:BuildingSpace` and `s4bldg:AirToAirHeatRecovery` instances. `Schedule` is a generic class that outputs predetermined schedules for a specific period, e.g. daily occupancy or setpoint profiles.

With the inputs and outputs defined for all components in the system, the three algorithms from section 5 can be applied to simulate the system. An overview of this workflow is shown in Fig. 6. As a first step, Algorithm 1 is applied to the set of components H, to identify and remove cycles caused by the controllers. This results in the updated set H^*, where four edges caused by the two controllers are removed. As shown in Fig. 6, all four removed edges point to the `s4bldg:BuildingSpace` instance, which holds the controlled properties. The obtained set of components H^* is then used as input for Algorithm 2 from which a sequence L is obtained, containing the components of H in a topologically sorted order. This sequence is then used in Algorithm 3 for simulating the original system H. To run this simulation, parameters have been specified for each model as listed in Table B.2. The parameters are based on a classroom in an actual university building. This is the case for the temperature prediction model for the `s4bldg:BuildingSpace` instance, which was trained on actual data collected from the classroom as described in our previous work [62,63]. For other components such as the `s4bldg:Fan` and `s4bldg:AirToAirHeatRecovery` instances, standard values, suitable for the demonstration case, are used. It should be stressed that the simulation is not intended to validate the accuracy of the presented component models, but rather to provide a proof-of-concept and validation of the overall modeling framework, information carryover, algorithms, and workflow.

6.3. Simulation results

For simulation, a 4-day winter period using actual historical weather data from Denmark is considered. The results for some of the components are shown in Fig. 7 where the output of each model is shown with the black dashed line while the colored lines show model inputs. The weather data used for the simulation are shown in Fig. 7i as a separate plot, characterized by low solar irradiance and temperatures.

Results for the temperature and CO2 models attached to the `s4bldg:BuildingSpace` instance are shown in the two top plots, Fig. 7a, and Fig. 7b. Fig. 7c shows the inputs and outputs of the temperature `s4bldg:Controller` instance. As shown, a constant 23 °C setpoint $T_{z,set}$ is provided as input to the `s4bldg:Controller` model during the whole period, resulting in a fluctuating valve position, which closes when the indoor temperature T_z exceeds $T_{z,set}$ and opens when the temperature drops below $T_{z,set}$. From Fig. 7e, the resulting heat consumption $\dot{Q}_h$ is shown for the `s4bldg:SpaceHeater` instance following the same fluctuating pattern, based on the water massflow $\dot{m}_w$.

Due to the low solar irradiation in the winter period, the shading position u_s has a limited effect on the indoor temperature and is therefore set to 0 during the entire period. However, as expected, running simulations for summer periods, it is observed that shade position has a high impact on indoor temperature, due to the higher irradiation levels.

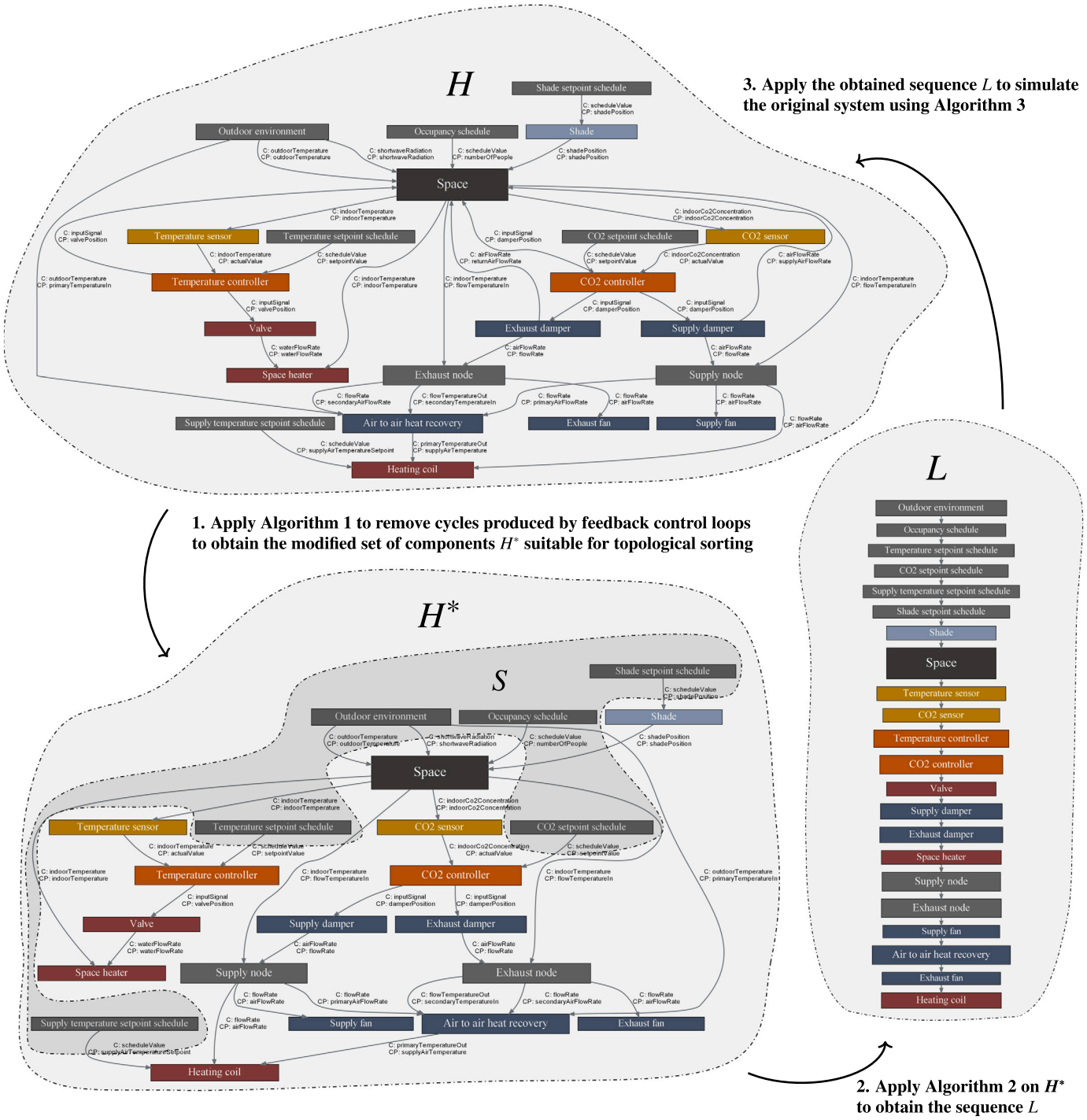

Fig. 6. Overview of Algorithm 1, Algorithm 2, and Algorithm 3 applied to the demonstration case.

Fig. 7b shows the simulation results for the CO_2 model attached to the `s4bldg:BuildingSpace` instance, where an artificially generated occupancy profile N_{occ} is used as input. Here, $\dot{m}_a$ represents both the supply and exhaust airflow as they are equal. Fig. 7d shows the CO_2 setpoint $C_{z,set}$ is set constant at 600 ppm, which the controller tracks by inversely actuating the supply and exhaust damper positions u_d. As shown in Fig. 7b, the predicted CO_2-concentration C_z is sensitive to rapid changes in occupancy and airflow, causing the tracking error to occasionally approach 100 ppm during occupancy periods.

The predicted power consumption $\dot{W}_{fan}$ for the supply `s4bldg:Fan` instance is shown in Fig. 7f. As expected, this consumption closely follows the supply airflow $\dot{m}_a$. The exhaust fan power consumption is identical due to identical models and balanced supply and exhaust airflows and is therefore not shown.

Fig. 7g shows the predicted outlet temperature at the supply side $T_{a,sup,out}$ of the `s4bldg:AirToAirHeatExchanger` instance. The inputs here are inlet temperature $T_{a,sup,in}$, i.e. the outdoor temperature, the exhaust inlet temperature $T_{a,exh,in}$, i.e. the indoor air temperature predicted by the `s4bldg:BuildingSpace` model, and the airflows at the supply and exhaust side $\dot{m}_a$.

The simulated heat consumption for the heating `s4bldg:Coil` instance $\dot{Q}_{hc}$ is shown in Fig. 7h. The consumption closely follows the trend of the supply airflow $\dot{m}_a$, due to a constant setpoint $T_{a,set}$ and a relatively invariant inlet air temperature $T_{a,in}$ between 18-20 °C.

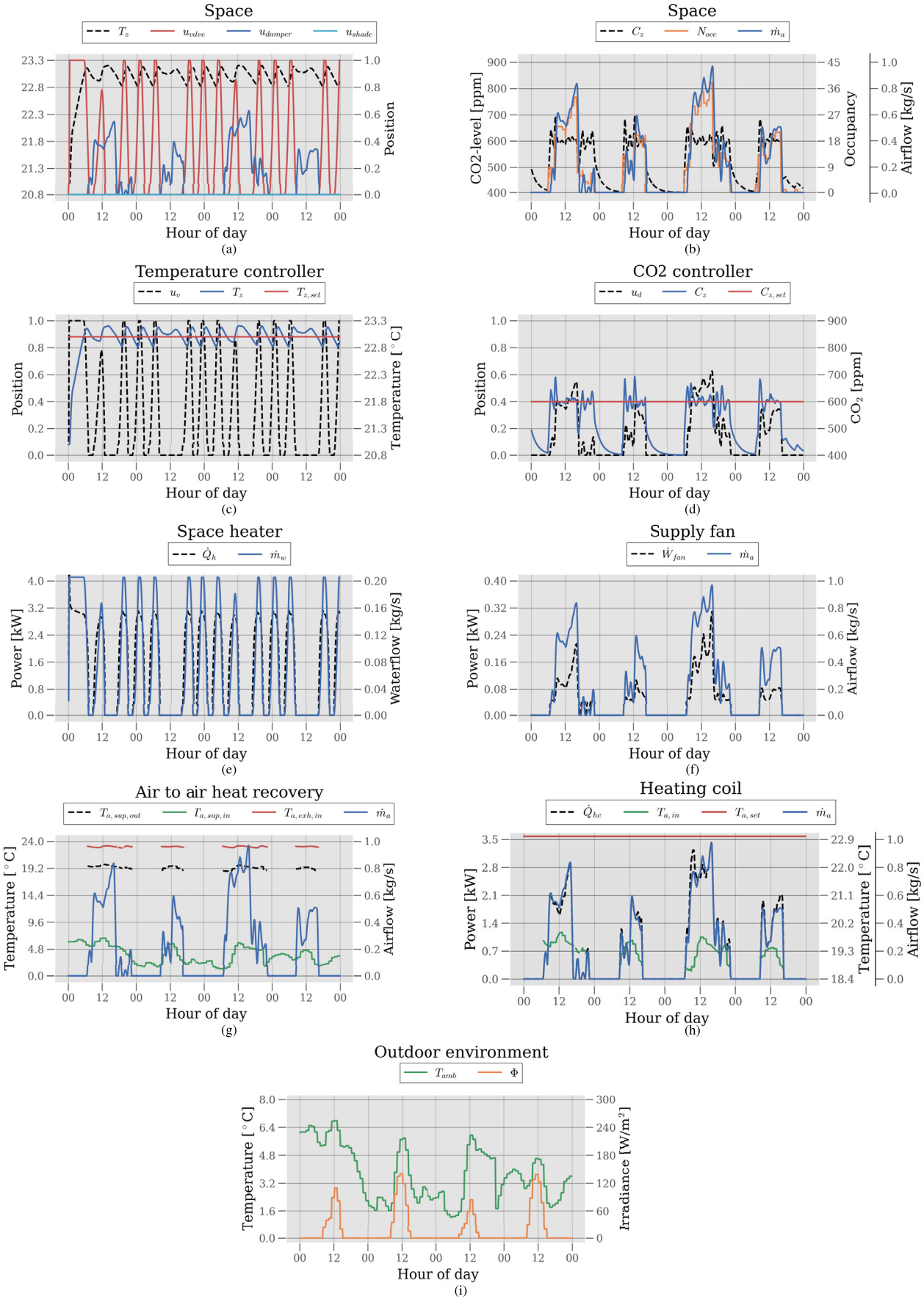

Fig. 7. Results of a 4-day winter period simulation for the demonstration case, employing the presented component models and modeling framework. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

6.4. Discussion

As demonstrated from the generated results, the modeling framework enables direct dynamic simulation of the components and systems represented through SAREF semantics. Although the demonstration takes basis in a simple single-zone system, the framework is not case-specific and can readily be applied to larger systems.

The majority of the presented component models are validated through other studies and implemented in simulation tools. However, their performance and effectiveness, as part of the presented modeling framework and a dynamic DT environment, are important to establish. Implementation of the framework in a real building case study is therefore planned as part of the next steps. Here, model parameters should be specified through a combination of available design data, and parameter estimation using measured operational data. The semantic alignment between SAREF4BLDG and IFC potentially provides an advantage in the task of obtaining the required design information and properties of the modeled devices and systems.

In addition, given the fact that sensor and meter equipment is virtually represented through the `saref:Sensor` and `saref:Meter` classes, the task of retrieving and comparing actual readings from the physical system and simulated readings from the virtual system for the same device is conceptually straightforward, which facilitates the implementation of continuous commissioning and parameter estimation. Relating to the demonstration case, the virtual temperature `saref:Sensor` instance reads the simulated indoor temperature shown in Fig. 7a while the CO_2 `saref:Sensor` instance reads the simulated CO_2-concentration shown in Fig. 7b. These readings are logged as `saref:Measurement` instances and retrieved through the `saref:hasFunction` and `saref:hasSensingRange` properties. Therefore, if actual readings from the two physical sensors are represented using the same concepts, the primary communication point from the physical system to the virtual system would thus pass through the physical and virtual `saref:Sensor` and `saref:Meter` instances. Here, robust and automated live communication with the physical IoT devices and preprocessing of the data will play a crucial role in the overall DT solution for proper use in continuous commissioning and parameter estimation.

7. Conclusion and future work

As part of the global energy transition, increasing the smartness of buildings is a key milestone, improving automation, efficiency, and flexibility. As part of this transformation, increased availability, sharing, management, and use of data from IoT devices and sensor networks both entail new opportunities as well as challenges.

Inspired by the demonstrated success and applications of the DT concept in other fields and industries undergoing a similar digital transformation as the building sector, the building DT is now emerging as a promising solution to handle and utilize these data streams for different important services such as fault detection, operational optimization, and strategy planning.

One of the fundamental constituents of a building DT which enables this is a dynamic simulation model. However, to avoid manual and time-consuming modeling workflows, and to ensure the adaptability and scalability of building DTs, close integration between the simulation models and IoT-devices at the building site is essential.

This work presents an innovative energy modeling framework that builds directly upon the semantics of the SAREF ontology with an emphasis on modular data-driven models. With a basis in the SAREF4BLDG extension, different candidate models were considered as extensions for a selection of classes. Using the existing ontology patterns from the SAREF4SYST extension, a generic method for systematically linking model inputs and outputs between components was presented. To enable direct dynamic simulation of the different components and systems, three essential algorithms were then presented. As a proof-of-concept, the framework was applied to model and simulate a generic single-zone system with common building components and systems. The presented framework is considered the foundation for future work that will consider an actual building as a case study with the integration of actual devices, real-time sensor equipment, parameter estimation, and delivery of different services. This will demonstrate the feasibility of using the DT services to enhance the efficiency and smartness of buildings.

The use of SAREF ontology as a semantic backbone for the modeling framework improves interoperability and integration with the actual physical system and devices and avoids translation to simulation software. This potentially enables broader adoption and scalability of building DTs. In addition, using the multiple domains covered by the SAREF extensions such as industry, energy, water, smart cities, etc., the DT concept could potentially be expanded to consider streets, districts, and cities.

CRediT authorship contribution statement

Jakob Bjørnskov: Conceptualization, Methodology, Software, Formal analysis, Investigation, Writing - original draft, Writing - review & editing, Visualization, Project administration. **Muhyiddine Jradi**: Conceptualization, Methodology, Writing - review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Jakob Bjoernskov reports financial support was provided by Danish Energy Agency.

Data availability

Data will be made available on request.

Acknowledgements

This work is carried out under the 'Twin4Build: A holistic Digital Twin platform for decision-making support over the whole building life cycle' project, funded by the Danish Energy Agency under the Energy Technology Development and Demonstration Program (EUDP), ID number: 64021-1009.

Appendix A

A.1. Valve

The valve model is based on the model used by Zajic et al. [64], and is given by the equations (A.1)-(A.2).

$$\tilde{u}_v = \frac{u_v}{\sqrt{u_v^2 (1 - N_v) + N_v}} \tag{A.1}$$

$$\dot{m}_w = \tilde{u}\, \dot{m}_{w,max} \tag{A.2}$$

In equation (A.1), the effectively applied stem position $\tilde{u}$ is calculated based on the control input $u \in [0, 1]$, and the valve authority $N_v \in [0, 1]$. Following, the massflow can then be calculated in equation (A.2) using the maximum massflow $\dot{m}_{w,max}$.

A.2. Space heater

The space heater model is based on the energy balance equation (A.3).

$$C_r \frac{\mathrm{d}\bar{T}_r}{\mathrm{d}t} = \dot{m}_w c_{p,w} (T_{w,in} - T_{w,out}) - UA(\bar{T}_r - T_z) \tag{A.3}$$

equation (A.3) represents the energy balance of the space heater with the stored heat on the left-hand side and the ingoing and outgoing heat flows on the right-hand side. Here, $\bar{T}_r$ is the effective temperature of the space heater, $T_{w,in}$ is the water inlet temperature, and $T_{w,out}$ is the

water outlet temperature. C_r is the heat capacity of the space heater, $\dot{m}_w$ is the water massflow, $c_{p,w}$ is the specific heat capacity of water, UA is the heat transfer coefficient, and T_z is the room temperature.

Xu et al. [65] models the effective temperature $\bar{T}_r$ of a space heater as one lumped element governed by equation (A.3) with a varying non-linear UA based on flow conditions. Tahersima et al. [66] consider the space heater as multiple finite elements that each are governed by equation (A.3) with a constant UA. For simplicity, this work considers the space heater as one lumped element with a constant UA and $\bar{T}_r = T_{w,out}$ is assumed, i.e. all energy transferred to the room is driven by the outlet water temperature. Through Backward Euler discretization [67] of equation (A.3), an explicit expression for the effective space heater temperature $\bar{T}_{r,t}$ at time t can thus be obtained as shown in equation (A.4).

$$\bar{T}_{r,t} = \frac{\left(T_{w,in}\dot{m}_w c_{p,w} + UA\,\bar{T}_{z,t-1}\right)\Delta t + \bar{T}_{r,t-1}C_r}{\left(UA + \dot{m}_w c_{p,w}\right)\Delta t + C_r} \tag{A.4}$$

Here, Δt is the timestep and $\bar{T}_{r,t-1}$ is the effective space heater temperature from the previous timestep. Based on the obtained space heater temperature $\bar{T}_{r,t}$, the heat transferred to the space heater from the heating system can be calculated in equation (A.5).

$$\dot{Q}_h = \dot{m}_w c_{p,w}(T_{w,in} - T_{w,out}) \tag{A.5}$$

A.3. Heating coil

The heating coil model is based on a simple heat balance equation by considering the sensible heat transfer from the coil to the air and is given by equation (A.6).

$$\dot{Q}_{hc} = \begin{cases} \dot{m}_a c_{p,a}(T_{a,set} - T_{a,in}), & \text{if } T_{a,in} < T_{a,set} \\ 0, & \text{otherwise} \end{cases} \tag{A.6}$$

In equation (A.6), $\dot{m}_a$ is the massflow of air, $c_{p,a}$ is the specific heat capacity of air, $T_{a,set}$ is the temperature setpoint of the supply airflow, and $T_{a,in}$ is the inlet air temperature of the airflow with respect to the heating coil. The model assumes that the heating coil is ideal and can always meet the specified supply air setpoint.

A.4. Cooling coil

Similarly to the heating coil, the cooling coil model also is based on a simple heat balance equation by considering the sensible heat transfer from the air to the coil and is given by equation (A.7).

$$\dot{Q}_{cc} = \begin{cases} \dot{m}_a c_{p,a}(T_{a,in} - T_{a,set}), & \text{if } T_{a,in} > T_{a,set} \\ 0, & \text{otherwise} \end{cases} \tag{A.7}$$

In equation (A.7), $\dot{m}_a$ is the massflow of air, $c_{p,a}$ is the specific heat capacity of air, $T_{a,set}$ is the temperature setpoint of the supply airflow, and $T_{a,in}$ is the inlet air temperature of the airflow with respect to the cooling coil.

A.5. Air to air heat recovery

The heat recovery model is based on the EnergyPlus *Air System Air-To-Air Sensible and Latent Effectiveness Heat Exchanger model* [53] and is given by equations (A.8)-(A.15).

$$f_{flow} = \frac{\frac{1}{2}\left(\dot{m}_{a,sup} + \dot{m}_{a,exh}\right)}{\dot{m}_{a,max}} \tag{A.8}$$

$$\epsilon_{75\%} = \begin{cases} \epsilon_{75\%,h}, & \text{if } T_{a,sup,in} < T_{a,exh,in} \\ \epsilon_{75\%,c}, & \text{otherwise} \end{cases} \tag{A.9}$$

$$\epsilon_{100\%} = \begin{cases} \epsilon_{100\%,h}, & \text{if } T_{a,sup,in} < T_{a,exh,in} \\ \epsilon_{100\%,c}, & \text{otherwise} \end{cases} \tag{A.10}$$

$$\epsilon_s = \epsilon_{s,75\%} + \left(\epsilon_{s,100\%} - \epsilon_{s,75\%}\right)\frac{f_{flow} - 0.75}{1 - 0.75} \tag{A.11}$$

$$\dot{C}_{a,sup} = \dot{m}_{a,sup} c_{p,a} \tag{A.12}$$

$$\dot{C}_{a,exh} = \dot{m}_{a,exh} c_{p,a} \tag{A.13}$$

$$\dot{C}_{a,min} = \min\left(\dot{C}_{a,sup}, \dot{C}_{a,exh}\right) \tag{A.14}$$

$$T_{a,sup,out} = T_{OA} + \epsilon_s\left(\frac{\dot{C}_{a,min}}{\dot{C}_{a,sup}}\right)\left(T_{a,exh,in} - T_{OA}\right) \tag{A.15}$$

In equation (A.8) the average flow fraction f_{flow} is calculated based on the supply air massflow $\dot{m}_{sup}$, the exhaust air massflow $\dot{m}_{exh}$, and the maximum air massflow $\dot{m}_{max}$. The heat recovery model has two modes of operation, heating, and cooling, reflected by the conditional effectiveness parameters $\epsilon_{75\%,h}$, $\epsilon_{100\%,h}$, $\epsilon_{75\%,c}$, $\epsilon_{100\%,c}$ in equation (A.9) and equation (A.10). Here, $\epsilon_{75\%}$ and $\epsilon_{100\%}$ is the efficiency at 75% and 100% airflow, while c and h denotes cooling and heating mode, respectively. Based on these effectiveness values, the operating efficiency ϵ is calculated in equation (A.11) by linear interpolation.

$\dot{C}_{sup}$ is the heat capacity rate of the supply air and $\dot{C}_{exh}$ is the heat capacity rate of the exhaust air given by equation (A.12) and equation (A.13), respectively. Following, the minimum heat capacity rate $\dot{C}_{min}$ is calculated in equation (A.14), and finally, the outlet air temperature on the supply side $T_{a,sup,out}$ can be calculated in equation (A.15), where T_{OA} is the outdoor temperature. In cases where the heat exchanger can provide setpoint control, $T_{a,sup,out}$ is corrected with equation (A.16) to avoid excessive heating or cooling of the air beyond the setpoint temperature.

$$T_{a,sup,out} = \begin{cases} \min(T_{sup,out}, T_{a,set}), & \text{if } T_{a,sup,in} < T_{a,exh,in} \\ \max(T_{sup,out}, T_{a,set}), & \text{otherwise} \end{cases} \tag{A.16}$$

In a previous investigation, the authors have proposed and demonstrated the use of a data-driven approach to identify the parameters $\epsilon_{75\%,h}$ and $\epsilon_{100\%,h}$ through collected temperature data by formulating a least squares optimization problem [68]. The resulting model provided good prediction accuracy on the outlet air temperature compared with the measured data.

A.6. Fan

The fan model is based on the EnergyPlus *Variable Speed Fan Model* [53] and is given by equations (A.17)-(A.19).

$$f_{flow} = \frac{\dot{m}_a}{\dot{m}_{a,max}} \tag{A.17}$$

$$f_{pl} = c_1 + c_2\, f_{flow} + c_3\, f_{flow}^2 + c_4\, f_{flow}^3 \tag{A.18}$$

$$\dot{W}_{fan} = f_{pl}\dot{W}_{fan,max} \tag{A.19}$$

In equation (A.17), the flow fraction f_{flow} is first calculated based on the current massflow $\dot{m}_a$ and the maximum massflow $\dot{m}_{a,max}$. Following, in equation (A.18), the part load ratio f_{pl} is calculated, based on the flow fraction and the power coefficients c_1-c_4, where $c_1 + c_2 + c_3 + c_4 = 1$. In equation (A.19), the part load ratio is then used to calculate the fan power consumption $\dot{W}_{fan}$, based on the power consumption at maximum airflow $\dot{W}_{fan,max}$.

A.7. Damper

The damper model describes the airflow through the damper $\dot{m}_a$ as a function of the damper position u and the parameters a, b, and c. It is based on the model presented by Huang [69] and is given by equation (A.20).

$$\dot{m}_a = a\, e^{b\,u_d} + c \tag{A.20}$$

Constraints can be imposed on the parameters a, b, c in equation equation (A.20) such that $m_a = 0$ when $u_d = 0$ and $m_a = m_{a,max}$ when $u_d = 1$. Hence, if a is given, b and c are given by equations (A.21)-(A.22) to satisfy these two constraints.

$$c = -a \tag{A.21}$$

$$b = \ln\left(\frac{\dot{m}_{a,max} - c}{a}\right) \tag{A.22}$$

A.8. Controller

Properly designed and implemented Building Management Systems (BMS) are vital for the building to adapt and react effectively to internal and external dynamic changes such as occupancy and climatic conditions [70]. The BMS is typically connected with a set of controllers that control the operation of various building systems. Controllers can come in different forms, e.g. rule-based control, conventional feedback control, predictive control, etc. However, the general aim of controllers is to modulate system inputs such that a predefined goal is achieved. For conventional controllers, the goal is typically to drive a specific system property toward the desired setpoint value. One of the most fundamental and widely used controllers is the Proportional Integral Derivative (PID) controller [71]. In discrete time, the controller behavior is described by equations (A.23)-(A.24) [72].

$$e_t = y_{set} - y_{meas,t} \tag{A.23}$$

$$u_t = u_{t-1} + K_p(e_t - e_{t-1}) + K_i\Delta t + \frac{K_d}{\Delta t}\left(e_t - 2e_{t-1} + e_{t-2}\right) \tag{A.24}$$

Where e_t is the residual between the setpoint y_{set} and the measured value $y_{meas,t}$ at time t. In equation (A.24), the output signal of the controller u_t is calculated based on the previous signal u_{t-1}, the previous residuals e_{t-1}, e_{t-2}, the parameters K_p, K_i, K_d, and the timestep Δt. Changing the type of controller is simply a matter of adjusting the parameters K_p, K_i, K_d, i.e., a P controller can be modeled with $K_p \neq 0$ and K_i, $K_d = 0$ while a PI controller can be modeled with K_p, $K_i \neq 0$ and $K_d = 0$. Additionally, a reverse-acting PID controller can be modeled with K_p, K_i, $K_d < 0$.

A.9. Building space

Indoor comfort is the driving force of essentially all energy use in buildings. Most BMS have automated control of indoor temperature to follow specified setpoint schedules. Additionally, Demand Controlled Ventilation (DCV) is a common strategy to ensure that mechanical ventilation is only operating during occupancy presence in the building with the aim of lowering power consumption and ensuring proper Indoor Air Quality (IAQ). For this purpose, CO_2 concentration is predominately measured as an indicator of IAQ and is controlled to follow certain setpoint schedules, by modulating the ventilation airflows [73]. Therefore, the task of accurately modeling and predicting temperature and CO_2 concentration is essential to properly represent the actual building space in the building DT. Therefore, the space component must be capable of predicting both the dynamic temperature and CO_2 concentration response, depending on the operation of the HVAC system, weather conditions, and occupancy use. The dynamic CO_2 concentration in rooms is typically modeled with the fundamental mass balance given by equation (A.25) [38,70].

$$m_z\frac{\mathrm{d}C_z}{\mathrm{d}t} = C_{sup}\dot{m}_{a,sup} - C_z\dot{m}_{a,exh} + K_{occ}N_{occ} \tag{A.25}$$

Where, m_z is the mass of the air contained in the room, C_z is the room CO_2 concentration, and C_{sup} is the CO_2-concentration of the supply airflow, which can be assumed equal to the outdoor air CO_2-concentration level at 400 ppm [70,74]. K_{occ} is the rate of CO_2 mass generated per occupant and N_{occ} is the number of occupants in the room. Pantazaras et al. [70], estimated K_{occ}, while synthetic EnergyPlus data was used as input for air flows and occupancy. Macarulla et al. [38], extended equation (A.25) to form a stochastic differential equation. Different configurations were tested for the estimated parameters. However, for the best-performing model, the actual occupancy count was used as input, while K_{occ} and the air flows were estimated as constants.

However, similarly to the transformation from equation (A.3) to equation (A.4), an explicit expression can be obtained for the room CO_2-concentration with Backward Euler discretization of equation (A.25). The obtained expression is given by equation (A.26), where the timestep Δt has been introduced.

$$C_{z,t} = \frac{m_z C_{z,t-1} + C_{sup}\dot{m}_{a,sup}\Delta t + K_{occ}N_{occ}\Delta t}{m_z + \dot{m}_{a,exh}\Delta t} \tag{A.26}$$

Indoor temperature forecasting has been studied extensively using different modeling methods covering both white-box, gray-box, and black-box approaches. However, recently, black-box approaches in the form of ANNs have become increasingly popular. Specifically, to deal with the transient and dynamic nature of indoor temperature, a specific branch of ANNs called Recurrent Neural Networks (RNN) is typically used. Under this category, the Long Short-Term Memory (LSTM) architecture is often used, due to its gated structure, which rectifies certain undesirable traits of its predecessor, the vanilla RNN [75]. The adaptability and prediction accuracy of LSTM networks have been demonstrated across many disciplines [76], including for the task of indoor temperature forecasting [77,78].

We have previously developed and demonstrated the use of the LSTM architecture [62,63] for indoor temperature modeling of a large set of spaces in a case study building. Therefore, this black-box approach is also applied in this work, which also demonstrates the flexibility of the proposed framework and the ability to combine different modeling approaches in a unified manner. The overall model architecture is summarized by equations (A.27)-(A.31), employing two sequential LSTM networks A and B.

$$\mathcal{X}_{A,t-1} = \left(T_{z,t-1}, T_{o,t-1}, \Phi_{s,t-1}, u_{v,t-1}, u_{d,t-1}, u_{s,t-1}\right) \tag{A.27}$$

$$(c_{A,t}, h_{A,t}) = \mathrm{LSTM}_A\left(\mathcal{X}_{A,t-1}, c_{A,t-1}, h_{A,t-1}\right) \tag{A.28}$$

$$\mathcal{X}_{B,t-1} = h_{A,t} \tag{A.29}$$

$$(c_{B,t}, h_{B,t}) = \mathrm{LSTM}_B\left(\mathcal{X}_{B,t-1}, c_{B,t-1}, h_{B,t-1}\right) \tag{A.30}$$

$$\Delta T_{z,t} = h_{B,t} \tag{A.31}$$

equation (A.27) defines the input $\mathcal{X}_A$ for LSTM_A. In this case, the inputs include indoor temperature T_z, outdoor temperature T_o, shortwave irradiation Φ_s, space heater valve position u_v, damper opening position u_d, and shades opening position u_{sh}. These inputs were chosen considering the thermal heat balance of the modeled spaces, and commonly available data in buildings. However, the inputs should be modified based on available sensors and equipment installed for the modeled space.

In equation (A.28), the input $\mathcal{X}_{A,t-1}$ as well as the state vectors $c_{A,t-1}$ and $h_{A,t-1}$ are provided as input for LSTM_A, which computes the updated state vectors $c_{A,t}$ and $h_{A,t}$. equation (A.29) then defines the input $\mathcal{X}_{B,t-1}$ for LSTM_B as the updated hidden state $h_{A,t}$. Using this input $\mathcal{X}_{B,t-1}$ as well as the state vectors $c_{B,t-1}$ and $h_{B,t-1}$, equation (A.30) then computes the updated state vectors $c_{B,t}$ and $h_{B,t}$, where the hidden state $h_{B,t}$ is treated as prediction target $\Delta T_{z,t}$, as shown in equation (A.31). Here, $\Delta T_{z,t}$ is the indoor air temperature difference between the previous and current timestep. After training, the model is used in inference for indoor temperature forecasting through a closed-loop configuration, given by equation (A.32). Here, the indoor temperature of the next timestep is obtained by simply adding the predicted temperature change $\Delta T_{z,t}$ with the known indoor temperature of the previous timestep $T_{z,t-1}$. The newly obtained temperature $T_{z,t}$ can then be fed back as input in equation (A.27), closing the loop.

$$T_{z,t} = T_{z,t-1} + \Delta T_{z,t} \tag{A.32}$$

Appendix B

Table B.2
Overview of parameter and constant values used for the demonstration case.

Component	Parameters	Constants
Valve	$N_v = 0.8$ $\dot{m}_{w,max} = 0.21$ kg/s	
Space heater	$UA = 162$ W/K $C_r = 125000$ J/K	$c_{p,w} = 4180$ J/kg/K $\Delta t = 600$ s
Heating coil		$c_{p,a} = 1000$ J/kg/K
Air to air heat recovery [1]	$\epsilon_{75\%,h} = 0.79$ $\epsilon_{75\%,c} = 0.79$ $\epsilon_{100\%,h} = 0.73$ $\epsilon_{100\%,c} = 0.73$ $m_{a,max} = 1.7$ kg/s	$c_{p,a} = 1000$ J/kg/K
Supply Fan[2]	$c_1 = 0.027828$ $c_2 = 0.026583$ $c_3 = -0.087069$ $c_4 = 1.030920$ $\dot{m}_{a,max} = 1.7$ kg/s $\dot{W}_{fan,max} = 1500$ W	$\rho_a = 1.225$ kg/m^3
Supply Damper[3]	$a = 5$ $\dot{m}_{a,max} = 1.63$ kg/s	
Exhaust Damper[3]	$a = 5$ $\dot{m}_{a,max} = 1.63$ kg/s	
Temperature controller	$K_p = 0.05$ $K_i = 0.8$ $K_d = 0$	
CO2 controller	$K_p = -0.001$ $K_i = -0.001$ $K_d = 0$	
Space[4]	$K_{occ} = 8.316 \cdot 10^{-6}$ kg/s/person $m_z = 571.5$ kg [-]	$\Delta t = 600$ s

[1] Efficiency values $\epsilon_{75\%,h}$, $\epsilon_{75\%,c}$, $\epsilon_{100\%,h}$, $\epsilon_{100\%,c}$ was selected based on the AHRI directory of certified product performance for a specific product [79].
[2] Power coefficients c_1-c_4 was selected based on the single-zone default values provided by the ANSI/ASHRAE/IES standard 90.1 [80].
[3] equation (A.21) and equation (A.22) are used to calculate c and b.
[4] CO_2-generation per occupant K_{occ} obtained for a classroom from [81]. The LSTM model contains too many parameters to show.

Table B.3
A subset of class descriptions and related properties from the SAREF4BLDG ontology extension for components commonly considered in building energy modeling [44].

Component	Description	Properties
Valve	*A valve is used in a building services piping distribution system to control or modulate the flow of the fluid.*	`closeOffRating`[op] `flowCoefficient`[op] `size`[op] `testPressure`[op] `valveMechanism`[dp] `valveOperation`[dp] `valvePattern`[dp] `workingPressure`[op]
Space Heater	*Space heaters utilize a combination of radiation and/or natural convection using a heating source such as electricity, steam or hot water to heat a limited space or area. Examples of space heaters include radiators, convectors, baseboard and finned-tube heaters.*	`bodyMass`[op] `energySource`[dp] `heatTransferDimension`[dp] `heatTransferMedium`[dp] `numberOfPanels`[dp] `numberOfSections`[dp] `outputCapacity`[op] `placementType`[dp] `temperatureClassification`[dp] `thermalEfficiency`[op] `thermalMassHeatCapacity`[op]

(*continued on next page*)

Table B.3 (*continued*)

Component	Description	Properties
Coil	*A coil is a device used to provide heat transfer between non-mixing media. A common example is a cooling coil, which utilizes a finned coil in which circulates chilled water, antifreeze, or refrigerant that is used to remove heat from air moving across the surface of the coil. A coil may be used either for heating or cooling purposes by placing a series of tubes (the coil) carrying a heating or cooling fluid into an airstream. The coil may be constructed from tubes bundled in a serpentine form or from finned tubes that give a extended heat transfer surface.*	airFlowRateMax^{op} airFlowRateMin^{op} nominalLatentCapacity^{op} nominalSensibleCapacity^{op} nominalUa^{op} operationTemperatureMax^{op} operationTemperatureMin^{op} placementType^{dp}
Air To Air Heat Recovery	*An air-to-air heat recovery device employs a counter-flow heat exchanger between inbound and outbound air flow. It is typically used to transfer heat from warmer air in one chamber to cooler air in the second chamber (i.e., typically used to recover heat from the conditioned air being exhausted and the outside air being supplied to a building), resulting in energy savings from reduced heating (or cooling) requirements.*	hasDefrost^{dp} heatTransferTypeEnum^{dp} operationTemperatureMax^{op} operationTemperatureMin^{op} primaryAirFlowRateMax^{op} primaryAirFlowRateMin^{op} secondaryAirFlowRateMax^{op} secondaryAirFlowRateMin^{op}
Fan	*A fan is a device which imparts mechanical work on a gas. A typical usage of a fan is to induce airflow in a building services air distribution system.*	capacityControlType^{dp} motorDriveType^{dp} nominalAirFlowRate^{op} nominalPowerRate^{op} nominalRotationSpeed^{op} nominalStaticPressure^{op} nominalTotalPressure^{op} operationTemperatureMax^{op} operationTemperatureMin^{op} operationalRiterial^{op}
Damper	*A damper typically participates in an HVAC duct distribution system and is used to control or modulate the flow of air.*	airFlowRateMax^{op} bladeAction^{dp} bladeEdge^{dp} bladeShape^{dp} bladeThickness^{op} closeOffRating^{op} faceArea^{op} frameDepth^{op} frameThickness^{op} frameType^{dp} leakageFullyClosed^{op} nominalAirFlowRate^{op} numberOfBlades^{dp} openPressureDrop^{op} operation^{dp} operationTemperatureMax^{op} operationTemperatureMin^{op} orientation^{dp} temperatureRating^{op} workingPressureMax^{op}
Controller	*A controller is a device that monitors inputs and controls outputs within a building automation system. A controller may be physical (having placement within a spatial structure) or logical (a software interface or aggregated within a programmable physical controller).*	
Building Space	*An entity used to define the physical spaces of the building. A building space contains devices or building objects.*	contains^{op} hasSpace^{op} isSpaceOf^{op}

References

[1] K. Ullah, I. Lill, E. Witt, An Overview of Bim Adoption in the Construction Industry: Benefits and Barriers, 2019.

[2] C. Boje, A. Guerriero, S. Kubicki, Y. Rezgui, Towards a semantic construction digital twin: directions for future research, Autom. Constr. 114 (2020) 103179, https://doi.org/10.1016/j.autcon.2020.103179, https://www.sciencedirect.com/science/article/pii/S0926580519314785.

[3] E. Glaessgen, D. Stargel, The digital twin paradigm for future nasa and u.s. air force vehicles, 2012.

[4] J. Guo, Z. Lv, Application of digital twins in multiple fields, Multimed. Tools Appl. 81 (08.2022), https://doi.org/10.1007/s11042-022-12536-5.

[5] M. Grieves, Digital twin: Manufacturing excellence through virtual factory replication, 03.2015.

[6] F. Xiao, S. Wang, Progress and methodologies of lifecycle commissioning of hvac systems to enhance building sustainability, Renew. Sustain. Energy Rev. 13 (2009) 1144–1149.

[7] L. Wang, S. Greenberg, J. Fiegel, A. Rubalcava, S. Earni, X.R. Yin, S. Woodworth, J. Hernandez-Maldonado, Monitoring-based hvac commissioning of an existing office building for energy efficiency, Appl. Energy 102 (2013) 1382–1390.

[8] Z. O'Neill, X. Pang, M. Shashanka, P. Haves, T. Bailey, Model-based real-time whole building energy performance monitoring and diagnostics, J. Build. Perform. Simul. 7 (2014) 83–99.

[9] E. Markoska, M. Jradi, B. Jørgensen, Continuous commissioning of buildings: a case study of a campus building in Denmark, in: 2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 2016.

[10] M. Jradi, K. Arendt, F. Sangogboye, C. Mattera, E. Markoska, M. Kjærgaard, C. Veje, B. Jørgensen, Obepme: an online building energy performance monitoring and evaluation tool to reduce energy performance gaps, Energy Build. 166 (2018).

[11] C.K. Metallidou, K.E. Psannis, E.A. Egyptiadou, Energy efficiency in smart buildings: iot approaches, IEEE Access 8 (2020) 63679–63699, https://doi.org/10.1109/ACCESS.2020.2984461.

[12] The cost-effectiveness of continuous commissioning® over the past ten years, 01.2008.
[13] H.B. Gunay, M. Ouf, G. Newsham, W. O'Brien, Sensitivity analysis and optimization of building operations, Energy Build. 199 (2019) 164–175, https://doi.org/10.1016/j.enbuild.2019.06.048, https://www.sciencedirect.com/science/article/pii/S0378778818334479.
[14] M. Laura Daniele, R. Garcia-Castro, M. Lefrançois, M. Poveda-Villalon, Saref: the smart applications reference ontology, Available at https://saref.etsi.org/core/. Online.
[15] D. Crawley, L. Lawrie, F. Winkelmann, W. Buhl, Y. Huang, C. Pedersen, R. Strand, R. Liesen, D. Fisher, M. Witte, J. Glazer, Energyplus: creating a new-generation building energy simulation program, Energy Build. 33 (2001) 319–331.
[16] F. Winkelmann, B. Birdsall, W. Buhl, K. Ellington, A. Erdem, J. Hirsch, S. Gates, Doe-2 supplement: Version 2.1e.
[17] S. Klein, W. Beckman, Trnsys 16: a transient system simulation program: mathematical reference, in: TRNSYS 5, 2007, pp. 389–396.
[18] A. Reddy, Literature review on calibration of building energy simulation programs: uses, problems, procedure, uncertainty, and tools, ASHRAE Trans. 112 (2006) 226–240.
[19] J. Reynolds, M. Ahmad, Y. Rezgui, Holistic modelling techniques for the operational optimisation of multi-vector energy systems, Energy Build.
[20] Y. Chen, M. Guo, Z. Chen, Z. Chen, Y. Ji, Physical energy and data-driven models in building energy prediction: a review, Energy Rep. 8 (2022) 2656–2671.
[21] I.J. Ramaji, J. Messner, E. Mostavi, Ifc-based bim-to-bem model transformation, J. Comput. Civ. Eng. 34 (04.2020), https://doi.org/10.1061/(ASCE)CP.1943-5487.0000880.
[22] A. Andriamamonjy, D. Saelens, R. Klein, An automated ifc-based workflow for building energy performance simulation with modelica, Autom. Constr. 91 (2018) 166–181, https://doi.org/10.1016/j.autcon.2018.03.019, https://www.sciencedirect.com/science/article/pii/S0926580517308282.
[23] A. Andriamamonjy, Automated workflows for building design and operation using openbim and modelica, Ph.D. thesis, 10.13140/RG.2.2.31108.78729, 11.2018.
[24] H.-Y. Chong, C.-Y. Lee, X. Wang, A mixed review of the adoption of building information modelling (bim) for sustainability, J. Clean. Prod. 142 (2017) 4114–4126, https://doi.org/10.1016/j.jclepro.2016.09.222, https://www.sciencedirect.com/science/article/pii/S0959652616315748.
[25] F. Mtibaa, K.-K. Nguyen, M. Azam, A. Papachristou, J.-S. Venne, M. Cheriet, Lstm-based indoor air temperature prediction framework for hvac systems in smart buildings, Neural Comput. Appl.
[26] Z. Fang, N. Crimier, L. Scanu, A. Midelet, A. Alyafi, B. Delinchant, Multi-zone indoor temperature prediction with lstm-based sequence to sequence model, Energy Build.
[27] B. Dong, C. Cao, S.E. Lee, Applying support vector machines to predict building energy consumption in tropical region, Energy Build. 37 (2005) 545–553.
[28] Z. Yu, F. Haghighat, B. Fung, H. Yoshino, A decision tree method for building energy demand modeling, Energy Build. 42 (2010) 1637–1646.
[29] S. Royer, S. Thil, T. Talbert, M. Polit, Black-box modeling of buildings thermal behavior using system identification, in: 19th IFAC World Congress, IFAC Proc. Vol. 47 (3) (2014) 10850–10855, https://doi.org/10.3182/20140824-6-ZA-1003.01519, https://www.sciencedirect.com/science/article/pii/S1474667016433392.
[30] P.P. Shinde, S. Shah, A review of machine learning and deep learning applications, IEEE, 2018, pp. 1–6.
[31] O.I. Abiodun, A. Jantan, A.E. Omolara, K.V. Dada, N.A. Mohamed, H. Arshad, State-of-the-art in artificial neural network applications: a survey, Heliyon 4 (11) (2018) e00938, https://doi.org/10.1016/j.heliyon.2018.e00938, https://www.sciencedirect.com/science/article/pii/S2405844018332067.
[32] S. Idowu, S. Saguna, C. Åhlund, O. Schelén, Applied machine learning: forecasting heat load in district heating system, Energy Build. 133 (2016) 478–488.
[33] Y. Guo, J. Wang, H. Chen, G. Li, J. Liu, C. Xu, R. Huang, Y. Huang, Machine learning-based thermal response time ahead energy demand prediction for building heating systems, Appl. Energy 221 (2018) 16–27.
[34] F. Pallonetto, C. Jin, E. Mangina, Forecast electricity demand in commercial building with machine learning models to enable demand response programs, Energy AI 7 (2022) 100–121.
[35] G. Pinto, Z. Wang, A. Roy, T. Hong, A. Capozzoli, Transfer learning for smart buildings: a critical review of algorithms, applications, and future perspectives, Adv. Appl. Energy 5 (2022) 100084, https://doi.org/10.1016/j.adapen.2022.100084, https://www.sciencedirect.com/science/article/pii/S2666792422000026.
[36] Z. Jiang, Y.M. Lee, Deep transfer learning for thermal dynamics modeling in smart buildings, in: 2019 IEEE International Conference on Big Data (Big Data), 2019, pp. 2033–2037.
[37] X. Fang, G. Gong, G. Li, L. Chun, W. Li, P. Peng, A hybrid deep transfer learning strategy for short term cross-building energy prediction, Energy 215 (2021) 119208, https://doi.org/10.1016/j.energy.2020.119208, https://www.sciencedirect.com/science/article/pii/S036054422032315X.
[38] M. Macarulla, M. Casals, M. Carnevali, N. Forcada, M. Gangolells, Modelling indoor air carbon dioxide concentration using grey-box models, Build. Environ. 117 (02.2017), https://doi.org/10.1016/j.buildenv.2017.02.022.
[39] X. Yu, L. Georges, M. Knudsen, I. Sartori, L. Imsland, Investigation of the Model Structure for low-Order Grey-Box Modelling of Residential Buildings, International Building Performance Simulation Association.
[40] H. Park, N. Martaj, M. Ruellan, R. Bennacer, E. Monmasson, Modeling of a building system and its parameter identification, J. Electr. Eng. Technol. 8 (2013) 975–983.
[41] A. Afram, F. Janabi-Sharifi, Gray-box modeling and validation of residential hvac system for control system design, Appl. Energy 137 (2015) 134–150, https://doi.org/10.1016/j.apenergy.2014.10.026, https://www.sciencedirect.com/science/article/pii/S0306261914010678.
[42] M. Pritoni, D. Paine, G. Fierro, C. Mosiman, M. Poplawski, A. Saha, J. Bender, J. Granderson, Metadata schemas and ontologies for building energy applications: a critical review and use case analysis, Energies 14 (2021) 2024, https://doi.org/10.3390/en14072024.
[43] L. Daniele, F. den Hartog, J. Roes, Created in close interaction with the industry: the smart appliances reference (saref) ontology, 2015, pp. 100–112.
[44] M. Poveda-Villalón, R. Garcia-Castro, Saref extension for building, Available at https://saref.etsi.org/saref4bldg/. Online.
[45] M. Poveda-Villalon, R. Garcia-Castro, P. Espinoza-Arias, Saref extension for smart city, Available at https://saref.etsi.org/saref4city/. Online.
[46] M. Lefrançois, Saref4syst: an extension of saref for typology of systems and their inter-connections, Available at https://saref.etsi.org/saref4syst/. Online.
[47] L. Daniele, Saref4ener: an extension of saref for the energy domain, Available at https://saref.etsi.org/saref4ener/. Online.
[48] International organization for standardization, iso 16739-1:2018. Available at https://www.iso.org/standard/70303.html. Online.
[49] P. Pauwels, W. Terkaj, ifcowl ontology, Available at https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2_TC1/OWL. Online.
[50] M.H. Rasmussen, M. Lefrançois, G. Schneider, P. Pauwels, Bot: the building topology ontology of the w3c linked building data group, Semant. Web (11.2020), https://doi.org/10.3233/SW-200385.
[51] P. Li, Z. O'Neill, J. Braun, Development of control-oriented models for model predictive control in buildings, ASHRAE Trans. 119.
[52] F. Jorissen, M. Wetter, L. Helsen, Simulation speed analysis and improvements of modelica models for building energy simulation, in: 11th International Modelica Conference, Versailles.
[53] U. Energy, Energyplus version 9.6.0 documentation.
[54] A. Ruano, E. Crispim, E. Conceição, M. Lúcio, Prediction of building's temperature using neural networks models, Energy Build. 38 (6) (2006) 682–694, https://doi.org/10.1016/j.enbuild.2005.09.007, https://www.sciencedirect.com/science/article/pii/S0378778805002136.
[55] T. Blochwitz, M. Otter, M. Arnold, C. Bausch, C. Clauß, H. Elmqvist, A. Junghanns, J. Mauss, M. Monteiro, T. Neidhold, D. Neumerkel, H. Olsson, J.-V. Peetz, S. Wolf, The functional mockup interface for tool independent exchange of simulation models, 2011, pp. 105–114.
[56] E.T.S. Institute, Smartm2m, extension to saref, part 3: building domain, etsi-ts-103 410-3, Available at https://www.iso.org/standard/70303.html. Online.
[57] C. Pang, J. Wang, Y. Cheng, H.L. Zhang, T. Li, Topological sorts on dags, Inf. Process. Lett. 115 (2015) 298–301, https://doi.org/10.1016/j.ipl.2014.09.031.
[58] R. Liu, A low complexity topological sorting algorithm for directed acyclic graph, Int. J. Mach. Learn. Comput. 4 (2014) 194–197, https://doi.org/10.7763/IJMLC.2014.V4.411.
[59] A.B. Kahn, Topological sorting of large networks, Commun. ACM 5 (1962) 558–562.
[60] J. Bjørnskov, Ontology-based building modeling framework, https://github.com/JBjoernskov/Twin4Build, 2023. (Accessed 23 March 2023).
[61] R. van der Weerdt, V. Boer, L. Daniele, B. Nouwt, Validating SAREF in a smart home environment, 2021, pp. 35–46.
[62] J. Bjørnskov, M. Jradi, C. Veje, A fully automated and scalable approach for indoor temperature forecasting in buildings using artificial neural networks, in: Building Simulation Applications BSA, 2022, pp. 349–356.
[63] J. Bjørnskov, M. Jradi, A modular thermal space coupling approach for indoor temperature forecasting using artificial neural networks, in: Building Simulation and Optimization BS0, 2022, https://publications.ibpsa.org/conference/paper/?id=bso2022_59.
[64] I. Zajic, T. Larkowski, M. Sumislawska, K. Burnham, D. Hill, Modelling of an air handling unit: a Hammerstein-bilinear model identification approach, in: International Conference on Systems Engineering, Las Vegas.
[65] B. Xu, L. Fu, H. Di, Dynamic simulation of space heating systems with radiators controlled by trvs in buildings, Energy Build. 40 (9) (2008) 1755–1764, https://doi.org/10.1016/j.enbuild.2008.03.004.
[66] F. Tahersima, J. Stoustrup, H. Rasmussen, P. Nielsen, Thermal analysis of an hvac system with trv controlled hydronic radiator, in: 2010 IEEE International Conference on Automation Science and Engineering, 2010, pp. 756–761.
[67] The backward Euler method, the trapezoidal method, John Wiley & Sons, Ltd, 2009, pp. 49–65. Ch. 4.
[68] J. Bjørnskov, M. Jradi, C. Veje, Component-level re-commissioning of a newly retrofitted Danish healthcare building, J. Build. Eng. 51 (2022) 104277, https://doi.org/10.1016/j.jobe.2022.104277, https://www.sciencedirect.com/science/article/pii/S235271022200290X.
[69] G. Huang, Model predictive control of vav zone thermal systems concerning bilinearity and gain nonlinearity, Control Eng. Pract. 19 (7) (2011) 700–710.
[70] A. Pantazaras, S.E. Lee, M. Santamouris, J. Yang, Predicting the co2 levels in buildings using deterministic and identified models, Energy Build. 127 (2016) 774–785, https://doi.org/10.1016/j.enbuild.2016.06.029, https://www.sciencedirect.com/science/article/pii/S0378778816305187.

[71] F. Belic, Z. Hocenski, D. Sliskovic, Hvac control methods - a review, https://doi.org/10.1109/ICSTCC.2015.7321372, 2015.
[72] S. Veeranna, U. Yaragatti, Analysis and implementation of discrete time pid controllers using fpga, Int. J. Electr. Comput. Eng. 2 (2010) 71–82.
[73] B. Merema, M. Delwati, M. Sourbron, H. Breesch, Demand controlled ventilation (dcv) in school and office buildings: lessons learnt from case studies, Energy Build. 172 (2018) 349–360, https://doi.org/10.1016/j.enbuild.2018.04.065, https://www.sciencedirect.com/science/article/pii/S0378778818301348.
[74] Chartered Institution of Building Services Engineers, Cibse guide a: Environmental design, chapter 4.2, Table 4.2, 2007.
[75] A. Sherstinsky, Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network, Physica D 404 (2020) 132306, https://doi.org/10.1016/j.physd.2019.132306, https://www.sciencedirect.com/science/article/pii/S0167278919305974.
[76] G. Van Houdt, C. Mosquera, G. Nápoles, A review on the long short-term memory model, Artif. Intell. Rev. 53 (12.2020), https://doi.org/10.1007/s10462-020-09838-1.
[77] F. Mtibaa, K.-K. Nguyen, M. Azam, A. Papachristou, J.-S. Venne, M. Cheriet, Lstm-based indoor air temperature prediction framework for hvac systems in smart buildings, Neural Comput. Appl. 32 (12.2020), https://doi.org/10.1007/s00521-020-04926-3.
[78] Z. Fang, N. Crimier, L. Scanu, A. Midelet, A. Alyafi, B. Delinchant, Multi-zone indoor temperature prediction with lstm-based sequence to sequence model, Energy Build. 245 (2021) 111053, https://doi.org/10.1016/j.enbuild.2021.111053, https://www.sciencedirect.com/science/article/pii/S0378778821003376.
[79] The Air-Conditioning, Heating, and refrigeration institute, directory of certified product performance, Available at https://www.ahridirectory.org/. Online.
[80] S. Goel, M.I. Rosenberg, Ansi/ashrae/ies standard 90.1-2010 performance rating method reference manual, 2016.
[81] A. Persily, L. De Jonge, Carbon dioxide generation rates from building occupants, Indoor Air 27 (03.2017), https://doi.org/10.1111/ina.12383.