BESMod - A Modelica Library providing Building Energy System Modules

Fabian Wüllhorst[1] Laura Maier[1] David Jansen[1] Larissa Kühn[1] Dominik Hering[1] Dirk Müller[1]

[1]Institute for Energy Efficient Buildings and Indoor Climate, E.ON Energy Research Center, RWTH Aachen University, Germany, fabian.wuellhorst@eonerc.rwth-aachen.de

Abstract

Towards the analysis and optimization of a coupled building energy sector, various component model libraries for hydraulic, ventilation, electrical, control, and building domains exist. However, no uniform open-source framework to couple these domains in a holistic building energy system simulation exists. Thus, we present *BESMod*, an open-source Modelica library, providing a modular approach towards domain-coupled building energy system simulations. *BESMod* relies on existing component specialized model libraries for the underlying physics. For the analysis of complex system simulations, user-friendly parameterization, consistent model interfaces, precalculated KPIs and debugging options are applied. The library is available at www.github.com/RWTH-EBC/BESMod. This paper motivates the library, lays out the interaction with existing model libraries and the general modular approach. An exemplary use case demonstrates the applicability of *BESMod*. Concluding, we motivate future development options.

Keywords: Coupled simulations, HVAC, Building, Modular

1 Introduction

Towards the integration of renewable energy into a domain-coupled building sector, coupled simulation based analysis of the hydraulic, ventilation, electric, control, and building domain can aid the development of innovative design and control methods (Ramsebner et al. 2021). To this extent, various Modelica libraries exist with the focus on modeling specific components from the domains hydraulic, ventilation, electric, and building (Modelica Association 2022). All existing libraries deliver relevant component[1] models. Besides providing simple system examples (cf. section 2), none of the mentioned open-source libraries aim for the aggregation of different components into compound energy systems.

One possible reason is the heterogeneity and complexity of energy systems. Especially for buildings, energy systems tend to be unique and, hence, not easily transferable. However, re-occurring subsystems[2] exist (EN 15316-1 2017). Researchers and practitioners often combine the same components to different hydraulic, ventilation, and electrical subsystems. For instance, the combination of a heat pump and a heating rod (hydraulic), a ventilator and a heat recovery (ventilation), or a photovoltaic power plant (PV) and a battery energy storage system (electrical) are typical subsystems. To enable the aggregation of components into subsystems and subsystems into a unique building energy system (BES), a high level of modularity of each subsystem is required (Baldwin and Clark 2000).

As no such library exists, researchers need to aggregate existing component models into a BES each time they want to model interactions between hydraulic, ventilation, electric, control and the building envelope. However, towards the integration of renewable energy and the coupling of sectors, analysis of these interactions are more important than ever (Ramsebner et al. 2021). Following this need, we highlight and close two research gaps:

- *Gap 1:* Modelica enables the modeling and coupling of different domains. While models for different domains exist, no uniform approach for coupling relevant domains within one BES exists.
- *Gap 2:* Coupled system models encompass large equation systems and countless parameters. Thus, the analysis is time-consuming and error-prone (Remmen et al. 2018). While existing libraries are user-friendly on component level, no clear approach exists for system analysis (cf. section 2).

To close these gaps, we develop the open-source library *BESMod*, a Modelica library for Building Energy System Modules. Regarding *Gap 1*, we realize a modular subsystem structure and a straightforward aggregation of subsystems into a unique BES. Further, we supply user-friendly and consistent approaches for parameterization, system analysis, and debugging to address *Gap 2*.

The remainder of this publication is structured as follows: Section 2 discusses and compares existing Modelica libraries for BES simulations in detail. Based on deducted

[1]In the context of this paper, a *component* refers to a single device in an energy system, such as a mover, a pipe or a storage.

[2]In this paper, we refer to subsystems as *modules*.

gaps, *BESMod* is presented in section 3. Section 4 describes an exemplary use case for the annual simulation of two building models. Finally, section 5 concludes with further development options and furture use cases.

2 Existing Libraries

In recent years, a variety of open-source libraries for the modeling language Modelica have been developed. Following the idea of *BESMod* library, we focus our state of the art on open-source Modelica libraries that are relevant for BES modeling and simulations. As a basis for our literature review, we take the Modelica Association's library list (Modelica Association 2022). Table 1 qualitatively categorizes libraries which are considered relevant for further evaluation and comparison with *BESMod* regarding the following criteria: On the component level, we distinguish between the existence, level of detail, and comprehensiveness of hydraulic (H), ventilation (V), electrical (E), control (C), and building (B) envelope models. Following *Gap 1*, we introduce the existence of system (Sys) configurations as an additional category. For instance, system packages include a connected overall BES or aggregate subsystems in a similar form. Furthermore, the libraries are evaluated regarding their modularity (Mod), i.e., the interconnectability of the subsystems. Following *Gap 2*, the aggregation of subsystems to a fully connected BES results in a tremendous increase in complexity. Therefore, simple and consistent parameterization (Par) is a key feature that modeling libraries should offer. For this category, the focus lies on consistency and simplicity of parameterization. For example, we evaluate if parameters are organized (e.g. in records), if parameter naming is consistent, whether the parameters were propagated and if measures to aggregate or simplify system parameters are offered. Finally, open-source development suffers from poor maintenance (Main) in many cases. Maintenance increases both model and library quality and is thus defined as the last category.

Twelve libraries are classified as relevant for *BESMod*. The first five libraries, namely *IBPSA* (Wetter, Blum, et al. 2019), *AixLib* (Müller et al. n.d.), *Buildings* (Wetter, Zuo, et al. 2014), *BuildingSystems* (Nytsch-Geusen et al. 2013), and *IDEAS* (Jorissen et al. 2018) are part of the cooperation within *IBPSA* Project 1 and are the follow-up projects of the former IEA EBC Annex 60 (Wetter and Treeck 2017; Wetter 2022). The aim of Annex 60 and Project 1 is to develop a new generation of open-source Modelica-based computational tools for building and district energy systems. One of the project's outcomes is the joint modeling library Modelica *IBPSA* as well as the four additional separate modeling libraries which all incorporate the *IBPSA* models but supplement them with further modeling efforts. In Table 1, we evaluate the libraries as they are, neglecting the fact that *AixLib*, *Buildings*, *BuildingSystems*, and *IDEAS* partially consist of *IBPSA* models and vice versa. For example, *AixLib*'s high representation of hydraulic models is partially due to a solid representation in the core library *IBPSA*. For instance, the *BuildingSystems* and *IDEAS* introduce e.g. battery and photovoltaic models or the *Buildings* adds detailed Building and EnergyPlus coupled building models.

For building envelope, the *IBPSA* offers the basis to model simplified building envelope models based on reduced-order approaches. These models are enhanced by the *Buildings* library which offers an extensive interface for EnergyPlus models and by the *AixLib*, introducing high-order modeling approaches (Xanthopoulou et al. 2021). System aggregation possibilities and modularity are similar among the regarded *IBPSA*-based libraries. All of them introduce examples to increase comprehensiveness or even integrate tutorials on how to build your own BES model, e.g. *IDEAS* and *Buildings*. Yet, system aggregation is not based on replaceable subsystems with uniform interfaces, leaving room for improvement. In addition, the coupling between electrical and thermal models is impeded by inconsistency, i.e. different electrical ports, or non-existent interfaces, i.e. electrical power as component output.

In addition to *IBPSA*-related libraries, few Modelica-based libraries focusing on holistic building performance simulation have successfully been introduced over the past years. Among them, the *BuildingSysPro* (Plessis, Kaemmerlen, and Lindsay 2014) is the only library outside the Project 1 cooperation (Wetter 2022) that also includes the *IBPSA*, but does not use it within the models. Like the *IBPSA*-based libraries, the *BuildingSysPro* includes models of all relevant subsystems, too. However, system aggregation is not included and the examples which refer to system simulation tend to focus on specific components. In addition, parameterization lacks the organization in records and detailed documentation, resulting in less manageable parameter structures. Furthermore, modularity is impeded due to the omission of consistent interfaces (e.g. busses) and subsystems, respectively.

Another library providing models for building performance simulations is the *FastBuildings* (Coninck et al. 2014). It is developed for low-order and grey-box modeling of buildings and simplified heating, ventilation, and air conditioning (HVAC) components for model predictive control applications. These individual subsystems can be integrated in other frameworks. However, the modeling and parameterization effort is rather high. Further, the last changes have been committed 7 years ago, thus the library appears to be inactive.

The *ThermofluidStream* modeling library enables detailed simulations of complex thermofluid systems and refrigerants (Zimmer 2020). Again, the individual subsystems are suitable to be incorporated in external BES simulation frameworks, but the library itself does not aim at providing them. The usage of different ports further inhibits an integration of *IBPSA*-based models.

The remaining open-source modeling libraries focus on selected domains rather than providing holistic coupled

Table 1. Comparison of open-source Modelica libraries offering models relevant to BES. The following abbreviations are used: H:=hydraulic, V:=ventilation, E:=electrical, C:=control, B:=building, Sys:=system, Mod:=modularity, Par:=parameterization, Main:=maintenance. The evaluation metrics are qualitative measures ranging from -- (poor representation/not existent), 0 (intermediate representation) to ++ (very good representation).

	BES subsystems									
Name	**H**	**V**	**E**	**C**	**B**	**Sys**	**Mod**	**Par**	**Main**	**URL**
IBPSA	0	0	--	-	0	0	+	0	++	https://github.com/ibpsa/modelica-ibpsa
AixLib	++	+	0	+	+	+	+	+	+	https://github.com/RWTH-EBC/AixLib
Buildings	++	+	+	+	++	+	+	+	++	https://github.com/lbl-srg/modelica-buildings
Building Systems	+	0	++	-	+	+	+	0	0	https://github.com/UdK-VPT/BuildingSystems
IDEAS	+	0	++	+	+	+	+	0	+	https://github.com/open-ideas/IDEAS
Building SysPro	0	0	+	+	+	-	--	-	0	https://github.com/EDF-Lab/BuildSysPro
FastBuildings	--	-	--	--	0	-	-	--	--	https://github.com/open-ideas/FastBuildings
Thermofluid-Stream	+	+	--	+	--	--	0	+	+	https://github.com/DLR-SR/ThermofluidStream
dhcSim	0	--	--	-	--	0	-	-	-	https://github.com/mabachmann/dhcSim
DisHeatLib	-	--	--	+	--	+	0	0	0	https://github.com/AIT-IES/DisHeatLib
TransiEnt	-	--	+	0	--	+	+	-	+	https://github.com/TransiEnt-official/transient-2.0.1
Building-ControlLib	--	--	--	+	--	--	--	-	--	https://github.com/TechnicalBuildingSystems/BuildingControlLib

building performance simulation. Nonetheless, they are potentially suitable to be integrated into the respective subsystem packages. For example, the *dhcSim* (Bachmann et al. 2021) and *DisHeatLib* (Leitner et al. 2019) both specialize on district heating and cooling simulations. Since they primarily aim is to enable large district simulations, the subsystems are simplified for low computational effort. *DisHeatLib* provides well documented and extensive system examples and basic control blocks for district use cases. However, the modularity is limited. *dhcSim* provides less extensive system examples and components. Another library that falls into the category of domain-specific libraries is the *TransiEnt* library (Andresen et al. 2015). Here, the focus lies on coupled energy systems on grid level. Consequently, the lower level BES is not part of the library. In addition, the library's documentation is limited and the parameterization effort is rather high. This impedes an integration into other frameworks. Nevertheless, the library provides extensive system examples. A satisfactory level of modularity is provided.

Finally, the *BuildingControlLib* is another domain-specific Modelica library (Schneider, Pessler, and Steiger 2017). The library aims at providing state-of-the-art control modules which occur in BES. Unfortunately, the library is not maintained anymore.

The comparison and categorization of different open-source Modelica libraries enabling building performance simulations reveals that the scientific community currently lacks a library that fulfills all of the following criteria:

- Selection of models for all relevant building subsystems, namely hydraulic, ventilation, electrical, control, and building envelope.
- Straightforward option to aggregate subsystems to coupled BES.
- Uniform interfaces and model structures among all subsystems to increase modularity and facilitate system aggregation.
- Consistent parameterization within and between the subsystems to reduce parameterization effort.

To bridge these gaps, we developed the modeling library *BESMod* which is presented in the following chapters. *BESMod* joins the rich pool of existing component models, providing models for full, domain-overarching BES simulations. Consequently, rather than being an alternative to the libraries discussed above, we enhance the current library portfolio and build upon the existing libraries by integrating them as dependencies.

3 Library Design

Based on the gaps identified in Section 1 and 2, we present the library design of *BESMod*. First, the integration of the existing component library pool is discussed (subsection 3.1). The realization of modularity is presented in subsection 3.2. The following subsections discuss the implementation of the subsystems hydraulic, ventilation, electrical, control, building, DHW and user profiles (subsections 3.3-3.9). After the description of the system aggregation in subsection 3.10, subsection 3.11 presents a keystone for high quality maintenance of *BESMod*, namely *Continuous Integration*.

Figure 1 illustrates *BESMod's* library structure. Besides the common `User's Guide`, the `Systems` package is the heart of *BESMod*. The `Utilities` package provides interfaces, icons and other. The `Examples` package contains the use case presented in section 4 as well as additional examples demonstrating the flexibility of *BESMod*. Last, the `Tutorial` package is an addition to the `User's Guide` to document how inheritance and replaceability of modules work.

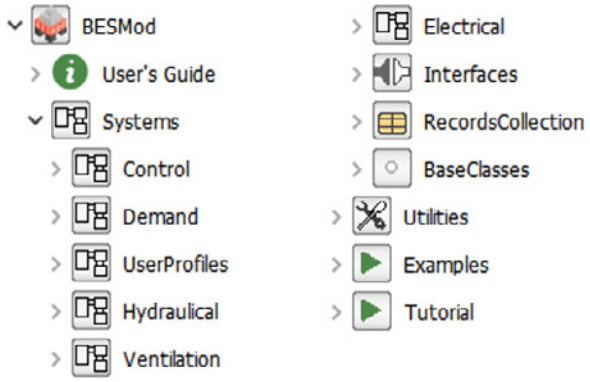

Figure 1. Package order of *BESMod*.

3.1 Dependencies

BESMod focuses on system models, not component models. As existing libraries already contain numerous component models of high quality, we build *BESMod* upon these libraries. Figure 2 depicts this setup. We distinguish between required and optional libraries.

The Modelica Standard Library (MSL) and the *IBPSA* library are required libraries. While the former is an obvious requirement, the choice of the *IBPSA* library is based on our state of the art. Out of all libraries listed in Table 1, the libraries derived from the *IBPSA* have the highest potential on component level for BES. Further, maintenance is applied, and the developer community is active. Thus, we choose the *IBPSA* library as the most suitable required library to integrate fluid models, media models, and other, which are compatible to the majority of component model libraries.

All other libraries are optional. At the current state of development, existing subsystems use the *AixLib*, the *Buildings*, and the *BuildingSystems* library.

The Python script `install_dependencies.py` installs all optional and required dependencies. For further information, the repositorie's `README.md` provides detailed documentation.
Regarding software dependency, *BESMod* is only tested using Dymola. As soon as the component libraries simulate in OpenModelica, *BESMod* will be tested for OpenModelica as well to enable the simulation of the open-source library in open-source software.

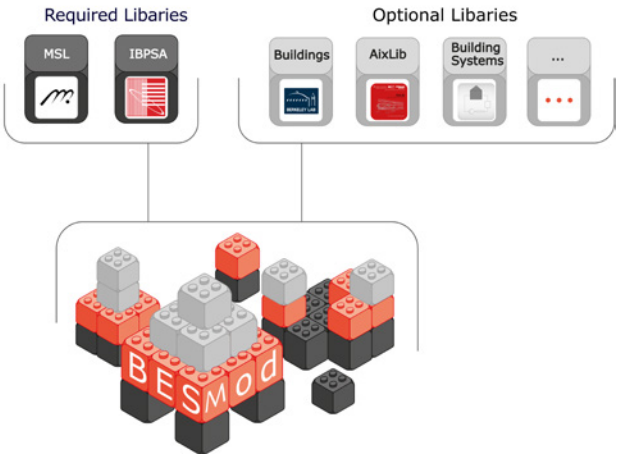

Figure 2. *BESMod* is built upon required and optional library dependencies.

3.2 Modular Subsystem Design

Baldwin and Clark (2000) define modules as "units in a larger system that are structurally independent of one another, but work together" (Baldwin and Clark 2000, p. 63). To achieve modularity, a framework needs to allow "for both independence of structure and integration of function" (Baldwin and Clark 2000, p. 63). Transfering these design rules to Modelica, we build each module with the same concept using expandable bus connectors, vector sized ports, and a uniform parameterization approach.

Expandable bus connectors: Each module is structurally independent. To exchange information with other modules, we implement expandable bus connectors (type `Real`, `Integer`, and `Boolean`). Five types of bus connectors arise. We distinguish between this variety of bus connectors by using color coding.

First, each module with HVAC components contains a bus to interact with the control module (white). As a naming convention, we apply the idea of the *AixLib*. For instance, a radiator transfer system receives the thermostatic valves set point `yValSet` as an input and sends the current valve position `yValMea` as an output. Using `Set` and `Mea` as indicators, users can always distinguish between control's and system's in- and outputs.

10.3384/ECP211869 DOI

Second, each module contains a bus to pass on relevant key performance indicators (KPI) of the module to the system (gray). This facilitates a fast analysis of complex BES. For instance, a generation subsystem using a heat pump may always output the consumed electrical energy, the supplied heat, and the number of starts. As we define these KPIs on a module level and propagate them using expandable bus connectors, the user can directly access the most important simulated states by analyzing the `outputs` bus.

The existing weather bus of the *IBPSA* library is used to distribute the boundary conditions to all subsystems (yellow). For user set points, all control models have access to the `UseProBus`, which includes for instance user presence and temperature set points (green). Last, the `BuiMeaBus` distributes current building measurements to the control subsystems (red).

Vector sized ports: Besides the exchange of control signals and system inputs, a module may exchange heat, mass, and electricity with other modules. While the usage of bus connectors for that purpose is generally possible, existing libraries commonly use the `FluidPort` for mass, the `HeatPort` for heat, and the `Pin` for electricity from the *MSL*. Thus, we consider the usage of expandable busses in this context as non-intuitive. Furthermore, a bus also only serves for information exchange in reality.

The individual use of ports depends on the type of subsystem and is elaborated in the accompanying subsections. Besides DHW supply, we apply vector sized ports. Thus, any number of parallel thermal zones, hydraulic circuits, or air ducts may be simulated. Again, this guarantees modularity.

Parameterization: Beutlich and Winkler (2021) motivate the decoupling of the "behavioral implementation from its actual design parameters" to facilitate a robust parameterization of simulation models. Applying this general method to *BESMod*, four model parameter types result.

1. **Top-Down**: All parameters given by the parent system. For instance, the heat demand of the building is a top-down parameter.

2. **Bottom-Up**: All parameters used by the parent system, but are inherently defined by the components in the subsystem. For instance, the system's pressure drop is given by the component models.

3. **Component choices**: All replaceable component models or packages. For instance, the hydraulic control system contains a replaceable thermostatic valve control. Thus, users may switch between P- and PI-controlled valves directly.

4. **Component records**: Each component in the system is equipped with a replaceable parameter record. This decouples the physical parameters from the modeling as motivated in (Beutlich and Winkler 2021). Ideally, such records would already exist in the component library. As most libraries do not support this approach, we add records for several components, for instance for the *IBPSA* mover models.

To enable a simple parameterization, composing the component records as a function of top-down parameters is convenient. Currently, the hydraulic subsystems follow the approach presented in previous work (Wüllhorst et al. 2022). In future versions, the electrical and ventilation subsystems may apply similar approaches.

While all subsystems follow this modular layout, differences occur depending on the subsystem's type.

3.3 Hydraulic Subsystem

In the following subsections, we highlight the choices for the **system layout**, the **connector** choice, the **parameterization** approach, and present already **available subsystems**.

System layout: The layout of the hydraulic subsystem follows (EN 15316-1 2017), which separates the system in generation, distribution, and transfer subsystems. To decouple the physical components from the control, we separate the control subsystem. The resulting subsystem layout is depicted in Figure 3 on the lower left.

Connectors: Water-based heating systems have two tasks. Firstly, they supply heat to possibly multiple thermal zones. Secondly, they provide DHW. Thus, the hydraulic subsystem contains heat and fluid ports from the MSL.

Parameterization: Wüllhorst et al. (2022) present a uniform method towards minimal parameterization effort in BES simulations. Inhere, each subsystem contains the exact same parameters to be quantified. The method follows the analogy of thermal and electrical systems and uses nominal heat and temperature demands to calculate remaining parameters. For instance, the mass flow rate is given by the nominal heat flow rate divided by the nominal temperature difference and the specific heat capacity of the fluid. As the demands are given by the building and DHW models, these are top-down parameters. Using correlations between top-down parameters and the selected component records, all remaining parameters are quantified. Fine-tuning is always possible, as we use the `final` modifier for top-down parameters only.

Available Subsystems: *BESMod* was originally developed for hydraulic-based BES simulations. Thus, a rich pool of subsystems already exist. Direct electric heating, bivalent heat pumps, gas boilers, solar thermal systems, different storage options, and radiator as well as underfloor heating subsystems may be combined. As control system, several approaches based on Vering et al. (2021) and the corresponding state of the art for the control of bivalent heat pump systems are implemented, too.

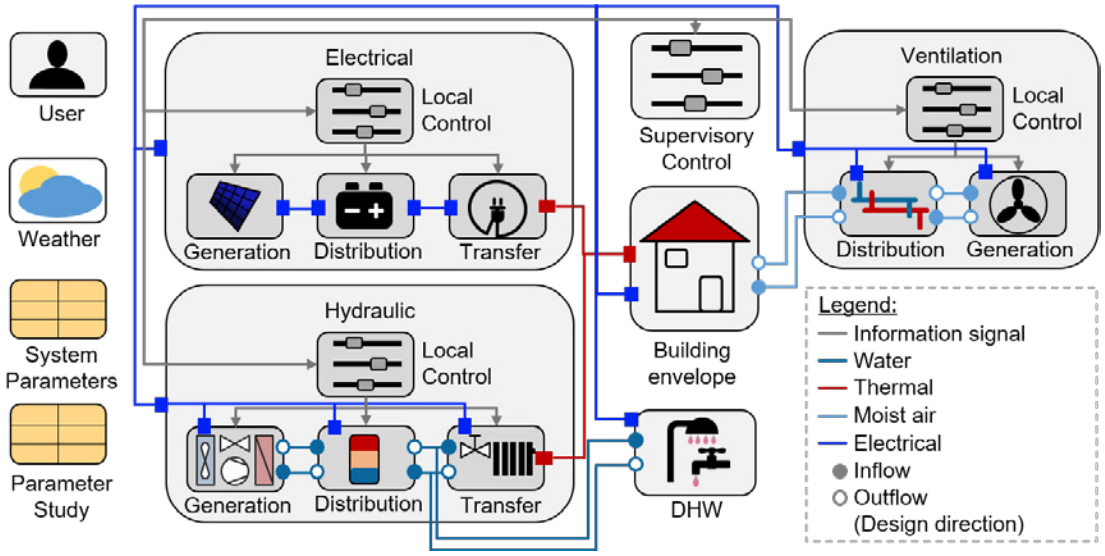

Figure 3. The aggregation of subsystems into the building energy system

3.4 Ventilation Subsystem

The ventilation system is used for heating, cooling, and air supply to control temperature, moisture and CO_2 concentrations. Especially towards the analysis and optimization of new buildings, simulation of ventilation coupled to hydraulics and electrics is vital.

System layout: As the hydraulic subsystem, the layout of the ventilation is based on the core idea of (EN 15316-1 2017). However, as the supply air directly flows into the thermal zones, no transfer system is required. The resulting subsystem layout is depicted in Figure 3 on the upper right.

Connectors: The ventilation system exchanges air with multiple thermal zones. Thus, fluid ports are used in addition to the usual bus connectors.

Parameterization: Similar to the hydraulic system, the parent system propagates temperature and heat demands top-down to each subsystem. Besides this propagation, no simplifying parameterization principles are applied. The supply flow rate has to be defined in each case. For future versions, we plan an automatic parameterization based on the net least area according to DIN 1946-6 (2019).

Available Subsystems: Currently, only a simple configuration exists in the open-source library. Inhere, a heat recovery system is simulated and equally distributed to all connected thermal zones. While only this example is open-source, more complex systems have been investigated in internal projects. For future versions, usage of air handling unit models, for instance, as presented in the *AixLib*, could be integrated.

3.5 Electrical Subsystem

The electrical subsystem in the *BESMod* library is used for the generation and distribution of electricity, as well as electricity-based space heating.

System layout: Based on this usage, the system layout is equal to the hydraulic system's one. Even though the decomposition into generation, distribution, and transfer derives from hydraulic systems, it is equally suitable for electrical systems.

Connectors: The existing *IBPSA*-related Modelica modeling libraries for BES focus on thermal and hydraulic subsystems and respective components. Even though the concept of electrical pins exists, e.g., in the MSL or the *Buildings*, they are not jointly used among the other libraries. In addition, most components which were especially modeled for thermal simulation assessments simplify electrical relations and solely calculate the electrical power output instead of detailed voltage or current relations. Hence, we introduce a simplified electrical connector which incorporates the flow variable power, only. Apart from that, the *IBPSA* weather bus and radiative and convective heat ports to the thermal zones are included, as well. The latter is an interface for electricity-based heat transfer by, e.g., electrical floor heating systems.

Parameterization: In contrast to the modeling approach by Wüllhorst et al. (2022), we parameterize each subsystem of the electrical system mostly independently. Nonetheless, the electrical subsystem predominantly relies on records and the definition of relations based on design parameters. For example, we introduce a design factor as a bottom-up parameter for system sizing relative to the building envelope's roof area, which is a top-down parameter.

Available Subsystems: The electrical subsystem incorporates the most commonly used electrical components in BES, namely a PV as generation subsystem and a battery energy storage subsystem to represent the distribution

side. The PV subsystem is based on the *AixLib* model. Following the library's aim of simple parameterization, the PV model's parameters are organized in records and system specific information is propagated to the top level. Again, to achieve a high degree of modularity, the system is introduced as a vector, providing the possibility to consider PV systems of different orientations, tilts, etc. Future developments for the generation models will include wind power plants.

As an exemplary model for the electrical distribution models, a simplified battery energy system based on the *BuildingSystems* library and a direct grid connection are included (Nytsch-Geusen et al. 2013). Again, different battery types can be selected based on a quick record change.

3.6 Building Envelope

The heart of any BES simulation is the model for the building envelope. For this subsystem, *BESMod* enables the connection to all three domains (hydraulic, ventilation, and electric) using heat, fluid, and electrical ports. The building subsystem outputs all relevant measurements to the other control systems. Currently, only temperature control (`TZoneMea`) is enabled. Future versions may add, e.g., moisture measurements and set-points.

For parameterization, no top-down parameters are relevant, as the building itself defines the dimensions of the HVAC components. Thus, various bottom-up parameters are used to propagate the building geometry to the residual subsystems. Currently, the building's height, ground area, roof area, and the area and height of each thermal zone have to be specified by the subsystem.

The current version contains two building models. The reduced order model used in TEASER (Remmen et al. 2018), as well as the detailed building envelope of the *Buildings* Library. The example is using the BESTEST validation model `Case600FF`.

3.7 DHW Subsystem

The DHW subsystem models the tapping of water from the hydraulic system. Thus, if the hydraulic system is not considered, DHW is disabled as well. For ports, fluid ports are an obvious choice.
Currently, DHW tapping according to *EN 16147* (2017) and Jordan and Vajen (2005) is included.

3.8 User Profiles

Both building and DHW subsystems greatly depend on the user profiles and vice versa. As no uniform approach to model internal gains, DHW tapping, etc. exists, no uniform user profile system is given. Instead, a replaceable model using the `UseProBus` enables a fully modular implementation of any user profile approach.
For instance, the `Case600FF` building model assumes area specific heat gains for internal gains. The `TEASER` model calculates the internal gains with relative inputs between 0 and 1. Besides internal gains, different set-points (constant, night setback, etc.) may be implemented.

While DHW tapping is a user action, we model the tapping profiles in the DHW subsystem itself. This avoids numerous user profile combinations, which would arise if different DHW and building model approaches are combined. Furthermore, disabling the DHW subsystem all together is straight forward if the profiles are in the DHW subsystem.

3.9 Control Subsystem

Last, a supervisory control system enables the simulation of an overarching home energy management system. Even though the hydraulic, ventilation, and electric subsystem all have a local control, no coordination of the coupled domains is possible. We consequently introduce the possibility to include a supervisory control model on system level.

The local control decides whether to enable supervisory control or not. Extending the supervisory control implemented by Blum et al. (2021), the user can select if the control signal derives from (i) the local control model only, (ii) a Modelica internal supervisory control model, or (iii) an external supervisory control as in BOPTEST. As we follow the BOPTEST concept for supervisory control, *BESMod* could be used as a BOPTEST test case in future versions (Blum et al. 2021).

For example, surplus PV electricity can be used for load to overcharge the thermal energy storage. While the basic control of the hydraulic system is modelled in the hydraulic subsystem's local control package, storage temperature set points are given by the supervisory control model on aggregated system level. Thus, is enabled and the coupling of domains is also realized from a control point of view.

3.10 System Aggregation

Figure 3 depicts the resulting system aggregation. To compose a custom BES, it is sufficient to extend the `PartialBuildingEnergySystem`, replace all gray subsystem models, and adjust the parameterization of the subsystems. The subsystem parameterization encompasses component choices and records. Fine-tuning is possible though adjustment of non-final bottom-up parameters.

Besides the replaceable subsystems, specification of weather boundary conditions and system parameterization is required.

For weather boundary conditions, we use the `ReaderTMY3`, which is compatible with building models based on the *IBPSA*. In the context of this study, no use case for replaceable weather models arose. However, future versions may include this, for instance, if weather data is not convertable to the TMY3 format.

The replaceable record `systemParameters` aggregates all top-down parameters, such as nominal outdoor air temperature or nominal room set temperature. The replaceable record `parameterStudy` collects parameters

users may want to perturb in a simulation study.

If the resulting system aggregation raises errors, we provide `Booleans` to disable the coupling of single subsystems. Besides debugging, these parameters may be used to discard single domains in an analysis. The only non-optional system is the building itself. Further, all subsystem packages contain dedicated testing models. With these tests, users can directly debug if an error is caused by the aggregation or a subsystem.

3.11 Continuous Integration

BESMod uses Continuous Integration (CI) to ensure the functionality and quality of the implemented models. The library is primarily hosted on GitHub and mirrored in the internal GitLab of RWTH Aachen University. Using the GitLab CI, a pipeline is triggered for each commit to the repository. The CI performs the following stages *Check* and *Simulate* using a Dymola Docker container.

The *Check* stage uses the built-in Dymola function to verify syntactic integrity and equal number of equations and variables. The *Simulate* stage searches for models in the library or the given Modelica package that extend `Modelica.Icons.Example` and simulates these models. Both stages are executed separately for the respective packages of the library. This procedure makes it easier to identify faulty models.

In the future, it is planned to include regression tests based on *BuildingyPy*[3] for individual integration tests of the overall systems.

4 Exemplary Use Case

As stated in section 1, user-friendly coupled BES simulations with integration of hydraulic, ventilation and electrical components are an open research gap. To illustrate the usability of *BESMod*, we aggregate an all-electric energy system for retrofit buildings.

4.1 System Layout

On the demand side, we demonstrate the flexibility and library independence of *BESMod* by using two different building model approaches. The first approach is the reduced order approach from the *AixLib* (Müller et al. n.d.). Second, the *Buildings* library `BESTEST` validation model `Case600FF` is used (Wetter, Zuo, et al. 2014). For DHW, we consider the tapping profile M according to *EN 16147* (2017). The applied weather conditions are taken from a test reference year of Aachen.

In the hydraulic subsystem, a bivalent heat pump system consisting of a pump, a heating rod, and a heat pump provides heat. In the distribution system, DHW is prioritized over space heating. To account for DHW loading phases, we consider a DHW and a space heating storage. For space heating, radiators transfer the heat to the thermal zones. Corresponding thermostats are controlled via PI controllers.

In the ventilation system, a heat exchanger recovers heat in the generation subsystem. This heat is distributed equally to all thermal zones.

In the electrical system, a PV system generates electricity. The PV area is designed based on the south facing roof area of the building and a design factor to specify the portion of the roof area to use for PV. Further, a lithium-ion battery based on manufacturer specifications maximizes self-consumption using the internal control logic presented by Nytsch-Geusen et al. (2013).

In the supervisory control system, a heuristic to overheat the DHW storage each day at 12 am for one hour is implemented. This control could increase the probability of an efficient heat pump operation using PV and higher ambient temperatures.

The resulting use case model is available in the librarie's Examples package as `UseCaseModelicaConferencePaper`. For specified parameters, records and structural parameter options, we refer to the source code. We highlight that further examples for different use cases based on previous work exist in the library as well (Vering et al. 2021; Wüllhorst et al. 2022).

4.2 Simulation Results

Performing an annual simulation[4] of the presented use case, we analyze the results for the two building models.

The focus of the analysis is not on a quantitative comparison of the two simulation models. Instead, we want to prove that models from different libraries can be coupled with little modeling effort and that the results can be analyzed quickly and easily on the basis of the integrated KPIs. For this purpose, excerpts of the KPIs calculated by *BESMod* for both building models (*AixLib* and *Buildings*) are shown in Table 2.

Table 2. Relevant KPIs for both building models.

KPI	*Unit*	*AixLib*	*Buildings*
$W_{\text{el,HP}}$	kWh	8,969	2,221
$W_{\text{el,HR}}$	kWh	36	440
$W_{\text{el,PV}}$	kWh	8,836	3,931
$Q_{\text{th,Tra}}$	kWh	25,456	5,011
$Q_{\text{th,DHW}}$	kWh	2,209	2,222
Discomfort	Kh	0.7	1.7
CPU time	s	326	357

Further, Figure 4 illustrates trajectories for all domains simulated in the use case. As the two buildings have a different geometry, inertia and insulation, the results differ regarding absolute and relative values. For a detailed understanding of the results, we invite the reader to analyze all KPIs, parameters and trajectories in the repository.

At this point, we want to emphasize that these results are calculated without additional adjustments to the mod-

[3] `https://github.com/lbl-srg/BuildingsPy`

[4] 600 s stepsize; Dassl Solver

10.3384/ECP211869 DOI

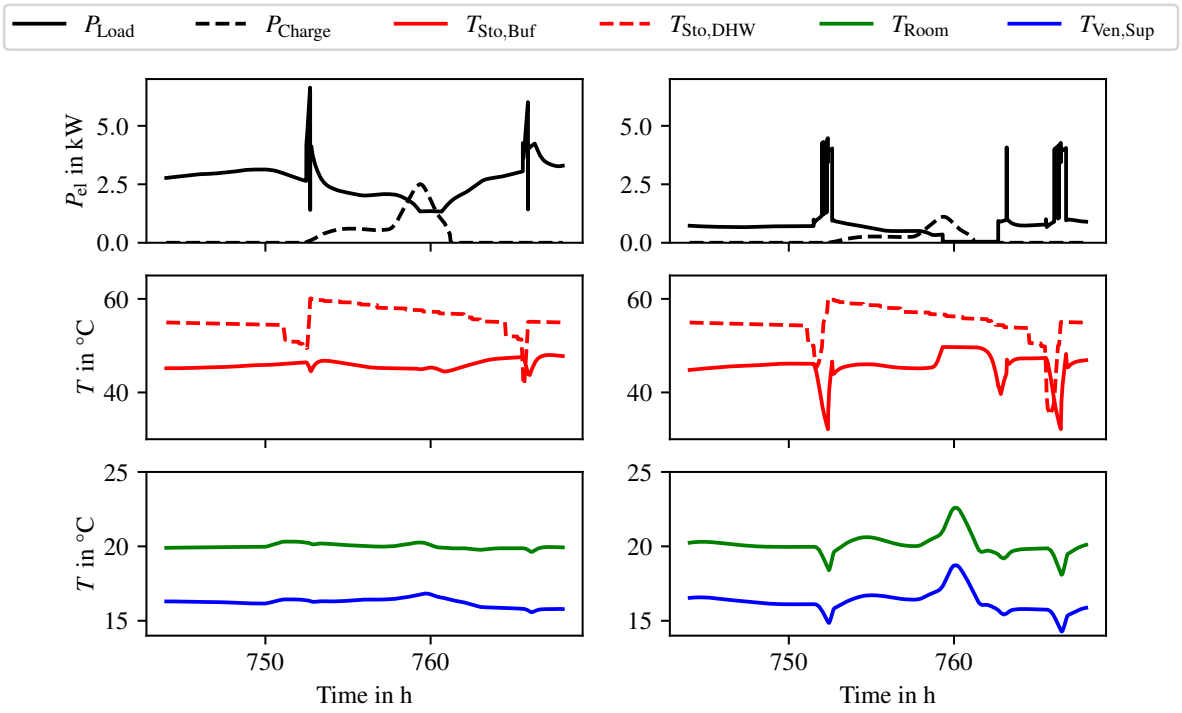

Figure 4. Relevant trajectories for the 1st of Feburary for the cases *AixLib* (left) and *Buildings* (right).

els. Further, only 184 lines of model code are necessary to build both examples using the newly presented *BESMod* library.

5 Conclusion

Closing the gap for a Modelica library which couples the domains hydraulic, ventilation, electrical, control, and building envelope in a modular, user-friendly manner, *BESMod* is presented. The use case demonstrates the ease of application, despite the high complexity of the domain-coupled building energy system simulation. Furthermore, in the face of the system's complexity, annual simulations take less than six minutes.

For the future library development, we hope modeling experts will contribute new modules and join the active developer community. As the current parameterization follows European guidelines, extensions towards international and American guidelines should be considered. Further, model validation is required as a next step. In here, *Continuous Integration* will integrate regression tests to ensure valid models for future developments. A shortcoming regarding modeling accuracy and the representation of real-world applications is that the electrical connectors are currently purely power-based. Future versions should add extensions towards current, voltage, and grid interactions. However, such development must be aligned with the underlying component libraries.

Regarding future use cases, the modular library approach lifts synergies for design and control domains.

For control, *BESMod* enables an efficient and realistic evaluation. Inhere, development of a test case for BOPTEST should be considered (Blum et al. 2021). Additionally, the modular system structure enables the efficient usage of ontologies such as Brick (Balaji et al. 2016). The modularity coupled to ontologies could enable the plug-and-play development of advanced control strategies.

For design, all domain-dependencies of renewable building energy systems are regarded in *BESMod*. Thus, simulation based design optimization methods as in (Vering et al. 2021) should be extended towards sustainable design strategies for building energy systems.

Acknowledgements

We gratefully acknowledge the financial support by the Federal Ministry for Economic Affairs and Climate Action (BMWK), promotional reference 03ET1495A.

This work emerged from the IBPSA Project 1, an international project conducted under the umbrella of the International Building Performance Simulation Association (IBPSA). Project 1 will develop and demonstrate a BIM/GIS and Modelica Framework for building and community energy system design and operation.

References

Andresen, Lisa et al. (2015). "Status of the TransiEnt Library: Transient Simulation of Coupled Energy Networks with High Share of Renewable Energy". In: *Proceedings of the 11th International Modelica Conference, Versailles,*

France, September 21-23, 2015. Linköping Electronic Conference Proceedings. Linköping University Electronic Press, pp. 695–705. DOI: 10.3384/ecp15118695.

Bachmann, Max et al. (2021). "dhcSim — A Modelica library for simple modeling of complex DHC systems". In: *Energy Reports* 7, pp. 294–303. ISSN: 23524847. DOI: 10.1016/j.egyr.2021.08.143.

Balaji, Bharathan et al. (2016). "Brick: Towards a unified metadata schema for buildings". In: *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments*, pp. 41–50.

Baldwin, Carliss Young and Kim B Clark (2000). *Design rules: The power of modularity*. Vol. 1. MIT press. ISBN: 9780262267649.

Beutlich, Thomas and Dietmar Winkler (2021). "Efficient Parameterization of Modelica Models". In: *Modelica Conferences*, pp. 141–146. DOI: https://doi.org/10.3384/ecp21181141.

Blum, David et al. (2021). "Building optimization testing framework (BOPTEST) for simulation-based benchmarking of control strategies in buildings". In: *Journal of Building Performance Simulation* 14.5, pp. 586–610.

Coninck, Roel de et al. (2014). "Grey-box Building Models for Model Order Reduction and Control". In: *Proceedings of the 10th International Modelica Conference, March 10-12, 2014, Lund, Sweden*. Linköping Electronic Conference Proceedings. Linköping University Electronic Press, pp. 657–666. DOI: 10.3384/ECP14096657.

DIN 1946-6 (2019-12-01). *Ventilation and air conditioning - Part 6: Ventilation for residential buildings - General requirements, requirements for design, construction, commissioning and handover as well as maintenance*. Tech. rep. Bruxelles, Belgium: CEN/TC 113.

EN 15316-1 (2017-09-01). *Energy performance of buildings - Method for calculation of system energy requirements and system efficiencies - Part 1: General and energy performance expression, Module M3-1, M3-4, M3-9, M8-1, M8-4; German version EN 15316-1:2017*. Tech. rep. Bruxelles, Belgium: CEN/TC 113.

EN 16147 (2017-01). *EN 16147:2017, Heat pumps with electrically driven compressors - Testing, performance rating and requirements for marking of domestic hot water units*. Beuth Verlag GmbH.

Jordan, Ulrike and Klaus Vajen (2005). "DHWcalc: Program to generate domestic hot water profiles with statistical means for user defined conditions". In: *Proceedings of the ISES Solar World Congress, Orlando, FL, USA*, pp. 8–12.

Jorissen, Filip et al. (2018). "Implementation and Verification of the IDEAS Building Energy Simulation Library". In: *Journal of Building Performance Simulation* 11 (6), pp. 669–688. DOI: 10.1080/19401493.2018.1428361.

Leitner, Benedikt et al. (2019). "A method for technical assessment of power-to-heat use cases to couple local district heating and electrical distribution grids". In: *Energy* 182, pp. 729–738. ISSN: 0360-5442. DOI: 10.1016/j.energy.2019.06.016.

Modelica Association (2022). *Modelica Libraries*. Last accessed: 25.04.2022. URL: https://modelica.org/libraries.html.

Müller, Dirk et al. (n.d.). "AixLib - An Open-Source Modelica Library within the IEA-EBC Annex 60 Framework". In: *Proceedings of the BauSIM 2016*, pp. 3–9. URL: http://www.iea-annex60.org/downloads/2016-bausim-aixlib.pdf.

Nytsch-Geusen, Christoph et al. (2013). "Modelica BuildingSystems – eine Modellbibliothek zur Simulation komplexer energietechnischer Gebäudesysteme". In: *Bauphysik* 35.1, pp. 21–29. ISSN: 01715445. DOI: 10.1002/bapi.201310045.

Plessis, Gilles, Aurelie Kaemmerlen, and Amy Lindsay (2014). "BuildSysPro: a Modelica library for modelling buildings and energy systems". In: *Proceedings of the 10th International Modelica Conference, March 10-12, 2014, Lund, Sweden*. Linköping Electronic Conference Proceedings. Linköping University Electronic Press, pp. 1161–1169. DOI: 10.3384/ECP140961161.

Ramsebner, Jasmine et al. (2021). "The sector coupling concept: A critical review". In: *Wiley Interdisciplinary Reviews: Energy and Environment* 10.4, e396.

Remmen, Peter et al. (2018). "TEASER: an open tool for urban energy modelling of building stocks". In: *Journal of Building Performance Simulation* 11.1, pp. 84–98. DOI: 10.1080/19401493.2017.1283539.

Schneider, Georg Ferdinand, Georg Ambrosius Pessler, and Simone Steiger (2017). "Modelling and Simulation of Standardised Control Functions from Building Automation". In: *Proceedings of the 12th International Modelica Conference, Prague, Czech Republic, May 15-17, 2017*. Linköping Electronic Conference Proceedings. Linköping University Electronic Press, pp. 209–218. DOI: 10.3384/ecp17132209.

Vering, Christian et al. (2021). "Towards an integrated design of heat pump systems: Application of process intensification using two-stage optimization". In: *Energy Conversion and Management* 250, p. 114888.

Wetter, Michael (2022). *IBPSA Project 1: BIM/GIS and Modelica Framework for building and community energy system design and operation*. Ed. by IBPSA. URL: https://ibpsa.github.io/project1/ (visited on 2022-04-26).

Wetter, Michael, David Blum, et al. (2019-05). *Modelica IBPSA Library v1*. DOI: 10.11578/dc.20190520.1. URL: https://www.osti.gov/biblio/1529269.

Wetter, Michael and Christoph van Treeck (2017-09). *IEA EBC Annex 60: New Generation Computing Tools for Building and Community Energy Systems*. ISBN: 978-0-692-89748-5. URL: https://www.iea-annex60.org/pubs.html.

Wetter, Michael, Wangda Zuo, et al. (2014). "Modelica Buildings library". In: *Journal of Building Performance Simulation* 7.4, pp. 253–270. ISSN: 1940-1493. DOI: 10.1080/19401493.2013.765506.

Wüllhorst, Fabian et al. (2022). "BEStPar: Towards Minimal Effort in Building Energy System Simulation Parameterization". In: *9th Conference of IBPSA Germany and Austria*.

Xanthopoulou, Konstantina et al. (2021). "Validation of a building model as part of the AixLib Modelica library for dynamic plant and building performance simulations". In: *Energy and Buildings* 250, p. 111248.

Zimmer, Dirk (2020). "Robust object-oriented formulation of directed thermofluid stream networks". In: *Mathematical and Computer Modelling of Dynamical Systems* 26.3, pp. 204–233. DOI: 10.1080/13873954.2020.1757726. eprint: https://doi.org/10.1080/13873954.2020.1757726.