15TH IEA HEAT PUMP CONFERENCE
MAY 26–29, 2026 | HOFBURG
VIENNA 2026
DECARBONISATION THROUGH INNOVATION

A holistic, simulation-based approach to integrating heat pumps into building energy systems

Ann-Kathrin Dreier*[a], Franziska Bockelmann[a], Jonas Hielscher[b], Andreas Schulte[b], Fynn Linnenbrügger[b], Wilhelm Tegethoff[b], Nicholas Lemke[b]

[a]siz energieplus, Hamburger Straße 277, Braunschweig 38114, Germany
[b]TU Braunschweig, Institut für Thermodynamik, Hans-Sommer-Str. 5, Braunschweig 38106, Germany

Abstract

Heat pump technology is a key driver in transforming the building heating sector. The TEN.efzn research project aims to design a novel, flexible plug & play heat pump module that is adaptable to various applications - from single-family houses to apartment buildings, in both centralized and decentralized setups, and across different climatic regions in Germany. To ensure robust design, simulation models are employed, offering efficient analysis under diverse boundary conditions.
A detailed heat pump model is required to represent real refrigeration circuit components and analyse thermodynamic behaviour under varying loads. Dymola, based on the Modelica language, provides an extensive library for this purpose. However, coupling weather-data-driven building simulations in Dymola is complex. TRNSYS, on the other hand, excels at generating load profiles for different building types and simulating complete energy systems, including heating circuits, storage tanks, and renewables. To leverage the strengths of both tools, they will be interconnected.
This coupling is achieved through a Functional Mock-up Unit (FMU), enabling time-synchronized co-simulation. At each time step, data such as mass flows, temperatures, and control signals are exchanged between TRNSYS and the FMU. Challenges arise in integrating the detailed FMU heat pump into TRNSYS, as additional input parameters and strict limit states complicate compatibility. Both programs must adapt to each other's characteristics, requiring iterative collaboration among research partners.

Keywords: Simulation models; interfaces; level of detail; overall view; interoperability

1. Introduction

As part of the TEN.efzn research project, a compact, modular plug & play heat pump with automotive components will be designed. One key aspect of the heat pump is its modular design, which enables the combination of several modules to achieve greater heating capacities thus serve as intended to be a solution for a wide variety of applications. Due to its compact design, it could be a suitable replacement for decentralized gas boilers in individual residential units and for central heaters in single-family and multi-family houses. Accordingly, it is important to design the heat pump module in such a way that it can reliably supply heat in combination with other modules or alone in a wide variety of applications.

The testing of the planned use cases exposed to several weather locations in a real test bench would exceed the available resources. Virtual simulations are appropriate solutions, representing complex interrelationships in a time and resource-efficient manner. Simulations used for example in the development process of a new product could expose it to various boundary conditions very easily [1]. Building and system simulations enable analysis of the operation of the heat pump module in single-family houses and apartment buildings, with several climates and with or without domestic hot water preparation.

For the new design of the heat pump module with the use of automotive components, in-depth research into the thermodynamics of the heat pump, the use of models for commercially available components, as well as the special development of control and optimization strategies under varying boundary conditions, coupled simulations of the heat pump with the buildings energy system are essential. The features of the heat pump and general design ideas are described in more detail in the IEA Heat Pump Conference 2026 paper "Compact, modular plug & play building heat pump using automotive components". In addition, further information on the use cases and the methodology for the development of a reference single-family and multi-family house can be found in the IEA Heat Pump Conference 2026 paper "Implementation and use cases of a compact, modular heat pump for residential buildings".

In order to achieve the goal of creating a compact, modular plug-and-play heat pump that utilises automotive components, a two-stage investigation will be conducted. In the first stage the coupling of different simulation tools specific to their specialization will be applied on a file-based manner. This simplified model will be used for static dimensioning and optimization of the heat pump components.

The second stage will utilize a direct dynamic coupling of the specialized simulation models utilizing a Functional Mock-Up Interface (FMI) for Co-Simulation inside TRNSYS. The picture (Figure 1) shows the links between different simulation programs and the responsibilities for the simulation programs by the institutions involved in the project. The advantages of cross-program simulations at this point are the strengths of each individual simulation program and the expertise of the institution behind it in dealing with the respective program. Sub-areas can thus be viewed and evaluated in different levels of detail.

The basis of the cross simulated heat pump module is the Dymola/ TIL Suite as shown in Figure 1. With the help of extensive component libraries, a model of the compact heat pump is to be developed in Dymola and tested in interaction with the other simulation interfaces. The Institute of Thermodynamics (IfT) at TU Braunschweig oversees this step. Based on the same program, the Institute for Solar Energy Research Hameln (ISFH) is investigating which errors could occur in the operation and control of the heat pump module and which solutions an AI-based approach could be used for fault detection and prevention. The Institute for Electrical Machines, Drives and Railways (IMAB) is responsible for the development of suitable power electronics for the heat pump module, which will also be incorporated into the overall model and later the physical test bench. The heat pump model, including control as well as AI-supported fault detection and optimization, is confronted with use cases in the TRNSYS simulation program. This transient simulation program provides the simulation of heat load coverage for space heating and/or domestic hot water for the predefined reference buildings and heating system variants. To consider use cases for the whole of Germany, 13 weather data profiles from 13 representative heat pump locations are used to generate load profiles. The aim is to investigate the integration of the heat pump module in a broader context of the energy systems as well as operating strategies and cascading, control and fault detection.

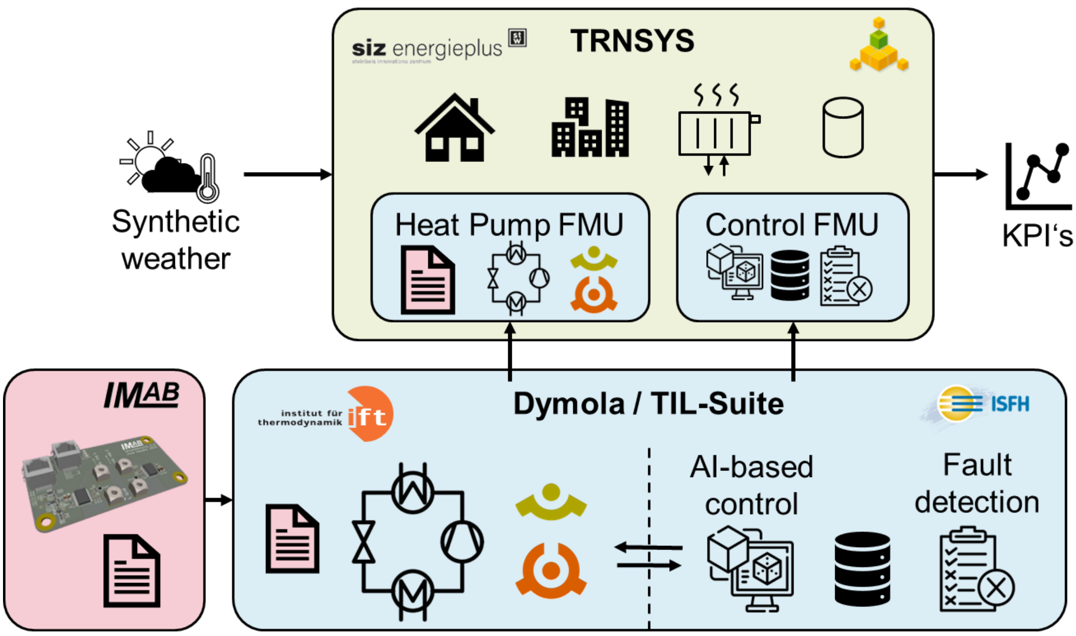

Figure 1. Overview of the simulations program coupling for the design of the heat pump module

In this paper the second stage of the model-based investigation and especially the coupling of the simulation tools *TRNSYS* and *Dymola* via a Functional Mock-up Unit (FMU) based on the Functional Mock-up Interface (FMI) standard will be discussed. Therefore, an interface for the heat pump FMU will be used for further usage inside the overarching model. The differences between a physics-based heat pump model and conventional heat pump model inside *TRNSYS* will be shown. The challenges that occur when coupling these two fundamentally different simulation tools will be addressed to derive design choices that facilitate easier integration into a *TRNSYS* deck for a typical building energy system.

2. Simulation tools and interfaces

In the following section the simulation tools used for modelling the building, the building energy system and the refrigeration cycle will be covered. The strengths and weaknesses as well as the abilities for coupling/ integrating of FMU's will be discussed.

2.1. TRNSYS

TRNSYS (*Transient System Simulation program*) is an open and modular simulation software for the simulation of transient simulation systems. The program was developed at the *University of Wisconsin-Madison* and has been providing a simulation platform primarily for the behavior of thermal and electrical systems for over 40 years [2] [3].The program consists of two areas. The first is the *Simulation Studio*, which provides a user interface for assembling and linking components in a modular way via input and output connections. The component models, called Types, are provided via libraries.

The TRNSYS installation package natively includes a standard library that provides a wide range of thermal, hydraulic and electrical components. Additionally, the *TESS library* extends the standard library by providing further Types for areas such as ventilation, solar applications and ground-coupled applications. The second part of TRNSYS, the computing core, the so-called Kernel, is responsible for solving the mathematical model with the help of the components and connections. The Kernel reads and processes the information stored in the model, solving it iteratively. It calculates convergence and outputs result variables. [3]

Another core point of TRNSYS is the multizone building simulation, which is executed by a separate upstream simulation program, *TRNBuild* (Type 56). This program can be used to simulate the thermal behavior of a building with user-defined geometries, building and construction standards, and radiation exchange models. The building geometry can be defined manually in TRNBuild or imported using three-dimensional building cubatures. The plugin "*Trnsys3d*" for *Trimble Sketchup* makes it possible to set cubatures, define zones and relate adjacent zones and shading cubatures to each other in a three-dimensional engineering program [4]. The specifications for the conditioning of the individual zones in terms of heating, cooling, mechanical and natural ventilation, as well as humidity, air exchange but also internal loads and gains and operation schedules can be determined for each defined zone separately. For thermodynamic components and material properties, TRNBuild can draw on extensive practice libraries and own components such as complex window models. Like the other regular Types, the TRNBuild model can be integrated into the parent model using inputs and outputs. The building directly interfaces with the dynamic system components of the model. The behavior of the coupled building and system can be modelled over time through direct information exchange. This makes it possible to map the technical building equipment, such as a heat pump, in more detail.[4]

The interaction between the various building technology components including heat pumps, buffer storage tanks and renewable energies can be shown in a simulation with a temporal resolution like it is given in TRNSYS. Control logics between the components can be implemented and checked. As a result, energy demands, and energy supply can then be compared and evaluated within the annual balance sheets. For the modelling of heat pumps, TRNSYS offers a selection of heat pump Types in both the Standard Library and the TESS Library, which can be used for different applications (air-to-water heat pump, air-to-air heat pump or brine-water heat pump or operating modes such as modulation or on/off mode).

For example, non-standard Type 402: *Compressor Heat Pump Including Modulation, and Frost Cycle Losses* requires two polynomials as input, representing the compressor and condenser performances at different temperature levels and different compressor speeds [5]. TESS-Type 927: *Normalized Water-to-Water Heat Pump*, on the other hand, requires an input data file for cooling operation as well as for heating operation, which contains the ratios between the thermal nominal power and achieved power at the operating point as well as the electrical nominal power and the achieved electrical load absorption at the operating point, depending on the mass flows and temperatures on the sources and sink side [6]. Most models of heat pumps are based on manufacturer-specific maps and the interpolations based on these. The only exception is Type 1327: Heat Pump Based on the *Lorenz Cycle*, in which the Lorenz cycle is described on the basis of user data and the heat pump performance is calculated [6].

Nevertheless, all TRNSYS heat pump Types have two things in common: they are described via algebraic equations only and are black boxes for the user. Apart from the defined outputs such as sources and sinks, temperatures, mass flows, energy absorption and heating rates as well as data on performance such as coefficient of performance (COP), TRNSYS cannot analyze or consider more detailed processes inside the heat pump and the work and interaction of the individual heat pump components such as compressors or evaporators. Additionally, the dynamics of the heat pump will not be covered. The transient mode-shifts and dynamic limitations of the specific heat pump will not be considered and the effect of different control strategies on the dynamic operation cannot be shown. Therefore, a more detailed physics based dynamic heat pump model is needed, which can be provided by *Dymola*.

2.2. Dymola

Dymola [7] is a simulation software based on the object-oriented equation based *Modelica* [8] modeling language. Dymola offers a variety of integrated advanced solvers with dynamic step size and variable order like *DASSL* [9] and *Cvode* [10] for efficient solving of stiff, hybrid differential algebraic equation systems which can contain discrete states. These types of equations are common when modeling physics-based thermodynamic cycles. The Modelica language enables the user to rapidly prototype and create new empirical and physics-based models while utilizing well known libraries like the *TIL-Suite* [11] from *TLK-Thermo GmbH* for modeling refrigeration/ heat pump cycles as well as other thermodynamic cycles and systems. These detailed models will enable the simulation-based research of operation strategies and Artificial Intelligence (AI)-based strategies, Fault Detection and Diagnostics (FDD), Fault Prevention (FP) and Condition Based Maintenance (CBM).
For investigation of the integration and control strategies of the heat pump system the model will need to be coupled with a building energy system. The following libraries are public available to integrate and couple a whole building system within Dymola: *Modelica Building Library* [12], *AixLib* [13], *BuildingSystems* [14] and *IDEAS* [15]. All of these libraries are based on the *Modelica IBPSA library* (International Building Performance Simulation Association) [16]. The Modelica IBPSA library is a free and open-source library with basic models for implementing a building and energy library with control systems. The IEA EBC Annex 60 [17] summarizes these software libraries based on Modelica modeling language, Functional Mockup Interface (FMI) [18] standard and Industry Foundation Classes (IFC) for building information modeling (BIM) [19] and other BIM related standards like Information Delivery Manual (IDM) and Model View Definitions (MVD).

For the BuildingSystems library a workflow of generating a building model from a BIM based CAD file (IFC2x3) was presented in [20] [21] and was continued within the *IBPSA project1* [22] until 2022. The generation of Modelica building models from an BIM based CAD file is limited to the building geometry. Air flow between the zones and configuration of internal loads must be parametrizes directly inside the BuildingSystems library. For generation of the BIM based CAD files additionally modeling software like *Autodesk Revit*, *Graphisoft ArchiCAD* or *Trimble SketchUp* with BIM export capabilities will be necessary.

Vering (2023) [23] presents a comprehensive optimal design method for heat pump systems, combining the heating and building systems at various levels of detail within Dymola. In addition to heuristic design methods, Vering also evaluates and optimizes simulation- and optimization-based design methods and summarizes them in published Modelica and Python models on GitHub. When simulating the building model, the AixLib library mentioned above is used by Vering. The models of the heat pumps are based on detailed modelling of the refrigerant circuits for the generation of heat pump characteristic curve, as well as the mapping of operational effects for refrigeration circuit and system control. [23]

Wüllhorst et.al. [24] show, that there are various advantages and disadvantages of the different open-source Modelica libraries relevant to building energy systems. Compared to the other building energy system libraries mentioned here and beyond, the AixLib only scores very good in the area of hydraulics. The other evaluated topics such as ventilation, electrical, control, building, system, modularity, parametrization or maintenance achieve only a medium to poor rating in Wüllhorst's comparative study. For this reason, a further development of the above-mentioned libraries was published in 2022.

BESMod [24] combines various building energy system libraries as necessary and optional components to form a uniform approach. Thus, the advantages of various existing Modelica libraries for the modeling of building energy systems on the above-mentioned evaluated topics are coupled in a modular, user-friendly manner. This shows that there are already promising solutions for the simulation of buildings and energy systems purely in Dymola/ Modelica. These libraries are open-source and therefore dynamically changing. In some places, practical application and validation are missing, as Wüllhorst et. al. [24] explain. Additionally, the interface for modeling a building system is quite complicated and the described tools are not yet widely spread outside of the research field. For these reasons a robust, validated simulation program that has been established outside the research society for building simulation and heat pump embedding was chosen for this project. TRNSYS based on *Fortran* offers a refined geometry and thermal behavior derivation from Intermediate Data Format (IDF-Files) and other quality of live features coupled with a wide range of validated and tested simulation models (see section 2.1 *TRNSYS*).

Dymola provides the capability of importing and exporting Functional Mock-up Units (FMU's) following the FMI standard by default and enables a simple integration with a multitude of simulation environments.

The heat pump therefore is fully modeled inside Dymola with components of the TIL-Suite consist of a heat pump cycle with propane as the refrigerant. As the components an efficiency-based compressor model, an offset-strip-fin heat exchanger as the condenser, an isenthalpic expansion valve and an offset-strip-fin heat exchanger for the evaporator are used. The heat pump cycle could easily be expanded, and components can simply be changed to its modular structure. First component models were fitted with measures of automotive components available from the spare parts market. The heat pump model uses continuous Proportional Integral (PI) controllers to control the heat sink outflow temperature (forward line to the heat load, building) and the superheating after the evaporator by adjusting the compressor speed and expansion valve opening. Additionally, a simple state machine is included to define the current state of the heat pump to enable a switching between on/off states. This physics-based heat pump model is integrated into a dedicated interface model providing the inputs and returning the outputs. The heat pump model included inside the interface model can be exported as a FMU for further integration into TRNSYS. Due to different units used inside the Dymola model and the TRNSYS model the unit conversion takes place inside the interface model inside Dymola. For further description of the modeled interface see Section 2.3

2.3. Traditional coupling of simulation tools and the FMI - Standard

In the present research project, the project partners have developed substantial expertise in TRNSYS for building simulation and Modelica for refrigeration cycle modeling. Consequently, a hybrid simulation approach by coupling both simulation environments is adopted in order to leverage this existing expert knowledge and to combine the strengths of both. Modelica in combination with the *TIL-Suite* model library, is employed to represent the heat pump in a high level of detail, thereby enabling the investigation of advanced research questions related to heat pump behavior within the context of the overall building energy system. TRNSYS, on the other hand, is a well-established simulation environment for building modeling and thus provides a complementary framework to Modelica.

Comprehensive simulation models of complex thermal systems can be implemented either as monolithic models within a suitable modeling language or by coupling individual sub models within the framework of a hybrid simulation [25]. Although both approaches are conceptually equivalent, they differ in terms of their respective strengths and limitations.

In the literature two principal approaches for implementing hybrid simulations can be identified. The first involves the coupling of a small number of simulation tools – typically two – via dedicated, tool-specific interfaces. The second comprises more general approaches that enable the coupling of multiple heterogeneous sub simulations. Examples for this kind of approach are the co-simulation architectures *TISC* [25]*, Ptolomy II* [26] and *AVL Model.CONNECT™* [27].

Given the inherent complexity of building energy system simulations and in line with common practice reported in the literature, this work adopts a hybrid simulation approach based on the coupling of two simulation tools in order to benefit from the respective specific strengths of both by using the open *Functional Mock-up Interface (FMI)* standard.

FMI is an open standard maintained as a Modelica Association Project. It defines an independent tool interface for exchanging simulation models and their information. Models are provided in the form of a compressed *Functional Mock-up Unit (FMU)* which contains model-specific metadata (modelDescription.xml), functionality data as C code or as a binary file (*.dll/*.so) and optional supplemental files such as documentation or parameter tables. [28]

To support a wide range of simulation workflows, the FMI-Standard defines two complementary export modes: *Model-Exchange* and *Co-Simulation*. In Model-Exchange, the FMU exposes the model equations and state variables, while the importing tool provides the numerical solver. The FMU offers functions for setting start values and inputs, but the importing tool determines time-step sizes and numerical solution schemes. This setup provides maximum flexibility and is particularly advantageous for small or moderately complex models. In contrast, Co-Simulation embeds the solver inside the FMU. The importing tool merely exchanges inputs and outputs at communication points, while the solver-specific decisions such as step size control remain within the FMU. This approach is well suited for complex models that rely on solvers optimized by the exporting tool and for scenarios where multiple simulation programs must be orchestrated with minimal solver-level coupling. [28]

Both approaches have distinct strengths and limitations. Model-Exchange offers direct access to the underlying mathematical model and allows adaptation to the solver settings of the importing tool. However, the performance for large models may suffer when optimized solvers are not available. Co-Simulation reduces flexibility but often achieves higher numerical robustness and execution speed. For integrating the heat pump FMU the Co-Simulation approach will be used to utilize a variable step size solver.

TRNSYS offers an open architecture that allows integrating user-developed components by implementing new Types in Fortran or *C++*. These Types extend the TRNSYS library and provide a mechanism to wrap external functionality. The mechanism has been used in previous work to integrate FMU-based Types for the Model Exchange approach [29]. In addition, TRNSYS supports the invocation of external programs like *CONTAM* or *Python* through dedicated Types [2]. For example, *Transsolar* provides Type 3157, which enables the execution of Python code via C Foreign Function Interface (CFFI). During the simulation, the TRNSYS DLL communicates with the CFFI layer, which turn interfaces with the Python runtime, allowing Python models to respond dynamically to simulation calls [30].

However, TRNSYS does not natively support FMI-based coupling. In particular, it lacks an Application Programming Interface (API) to parse FMU contents or manage FMI communication steps [31]. To overcome this limitation, Type 6139 was developed as an FMI adapter that implements a front-end/ back-end structure and mimics the functionality of an FMI-compliant API. Using this adapter as FMU export utility of TRNSYS-models, various Co-Simulation scenarios, such as TRNSYS coupled with Dymola or *Simulink*, have been demonstrated [17] [31] [32]. For the design of the plug & play heat pump in TEN.efzn, such a Co-Simulation setup between TRNSYS and Dymola is likewise required, but for the integration of FMUs into the TRNSYS Model. The approach of directly integrating the FMU as a TRNSYS type, as demonstrated in [29], will be chosen to improve handling of the FMU models. However, the Co-Simulation approach will be chosen instead, which requires the development of a new interface.

The associated project partner *TLK-Thermo GmbH* based in Brunswick provided us an experimental Co-Simulation based FMI-Adapter for converting FMU's into TRNSYS Types. This enables the integration of a dynamic physics-based heat pump model via a FMU directly into TRNSYS as a binary file.

3. Implementation of the coupling

3.1. Design of the interface

The heat pump is modelled as a Functional Mock-up Unit (FMU) in Dymola and is internally controlled with a dynamic integration step to resolve fast dynamics. In the TRNSYS environment the FMU is used as a black-box component that exchanges information only at the TRNSYS simulation step. The interface modeled for the heat pump within Dymola described below and shown in Figure 2 provides: selection of fluid Types for source and sink, harmonization of the disparate time steps, scalability to several parallel heat pump modules, a complete set of inputs, outputs and diagnostic signals required for system-level control.

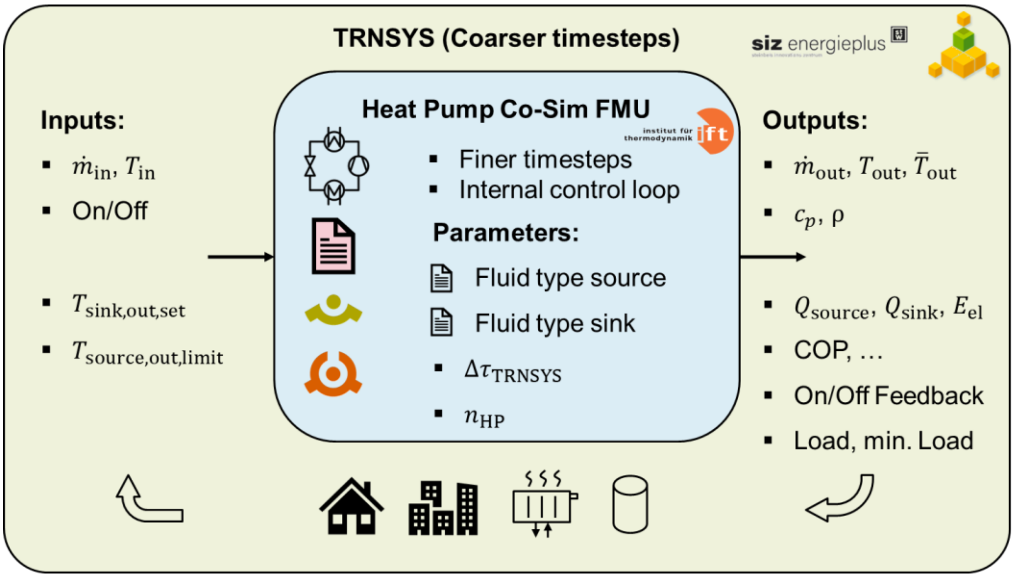

Figure 2. Heat Pump FMU-Interface integrated inside TRNSYS Simulation environment

The parameters for the fluid Types of the source and the sink are supplied during the FMU initialization. The FMU then loads the corresponding fluid Type for description of fluid behavior utilizing the *TILMedia Suite* [33] from *TLK-Thermo GmbH.* Additionally, the step size of TRNSYS $\Delta\tau_{\text{TRNSYS}}$ and the number of parallel heat pump modules n_{HP} will be passed to the FMU during initialization. The TRNSYS step size will be used to calculate the rolling-mean output temperatures $\overline{T}_{\text{out}}$ (Eq. 1) of the heat sink and source over each TRNSYS time step and is returned as an output. This mitigates integration error caused by the mismatch between the fine internal solver and the coarser TRNSYS discretisation.

$$Q = \int_{\tau}^{\tau+\Delta\tau_{\text{TRNSYS}}} \dot{Q}\,\mathrm{d}\tau = \dot{m} \cdot c_p \cdot \left(\overline{T}_{\text{out}} - T_{\text{in}}\right) \cdot \Delta\tau_{\text{TRNSYS}} \tag{1}$$

At the beginning of every timestep of the simulation several inputs will be passed to the heat pump FMU. An enabling signal provides supervisory on/off. From the heat source the mass flowrate $\dot{m}_{\text{source,in}}$ and the inlet temperature $T_{\text{source,in}}$ must be provided. A setpoint for the limitation of the source outflow temperature $T_{\text{source,out,limit}}$ can be set to limit the minimal source outflow temperature by reducing the load of the heat pump. This limiter ensures a minimal input temperature for earth heat sources if used. From the heat sink (return line of the heat load) the mass flowrate $\dot{m}_{sink,\text{in}}$ and the return temperature $T_{\text{sink,in}}$ must be provided. The setpoint for the sink outflow temperature $T_{\text{sink,out,set}}$ will drive the internal load control of the heat pump that continuously adjusts the compressor speed.

At the end of every time step the heat pump FMU provides several outputs. The corresponding outlet/ supply temperatures ($T_{\text{source,out}}$ and $T_{\text{sink,out}}$) and mass flowrates ($\dot{m}_{source,\text{out}}$ and $\dot{m}_{sink,\text{out}}$) of the heat source and sink are provided for connecting further Types to the energy system. The transferred heat of the source Q_{source} and sink Q_{sink} within a TRNSYS timestep calculated as an integral over the smaller internal time step will be provided. The rolling mean temperatures for the source ($\overline{T}_{\text{source,out}}$) and sink ($\overline{T}_{\text{sink,out}}$) outlet temperatures during a TRNSYS time step ensure that the net energy exchanged matches the energy implied between the different time scales of the FMU and TRNSYS. Several additional values like the fluid properties of source and sink, energy demand of the compressor E_{el} and the coefficient of performance (COP) will be available as outputs. The heat pump will provide feedback about the on/off status, the current and the minimal load. This information will be utilized for the integration of the heat pump into the control of the energy system.

The heat pump is integrated inside a TRNSYS deck with a building and an energy system. The inputs and outputs of the heat pump can be connected with other TRNSYS Types like earth heat exchangers, fluid pumps and thermal energy storage.

In contrast to the continuous simulation in Dymola with smaller time steps, the mathematical model in TRNSYS is solved within larger time steps using the successive substitution method [2]. This is a numerical method for iteratively solving equations with unknowns [34].

In terms of TRNSYS, the deck file is read out by the Kernel, the computing core, in the start time step. In the deck file, all Types and relationships of the Types with each other are stored. Each Type, on the other hand, represents a component such as a buffer tank for example and in turn consists of a mathematical model. Input and output connections thus create a complex mathematical system, which the Kernel tries to solve in each simulation time step related to the changing boundary conditions. Since the input values of some components are the output values of other components, the desired convergence (complete solution of the system of equations) is approached in the first iteration step based on random assumptions or predefined initial values. The second iteration step, in turn, continues its calculations based on the results of the first iteration step. This is repeated until the convergence of the system is achieved. Once this is done, the simulation time step is terminated. [2]

The calculated variables such as return, source- or sink mass flows, as well as temperatures or on/off signals are exchanged with the heat pump FMU and thus with the Dymola heat pump model. The output values of the Dymola model of the current time step, on the other hand, serve as input values for the TRNSYS model in the next time step, on the basis of which the convergence of the system is again iterated. TRNSYS exchanges information with the FMU interface only at the end of each time step. In each iteration step, it calculates the values transferred once from the FMU heat pump. Neither the TRNSYS model nor the Dymola model can therefore react to changing conditions or iteration intermediate steps with regard to heat pump operation during a TRNSYS time step.

3.2. Challenges

Heat pump models provided by TRNSYS (discussed in paragraph 2.1) and Dymola (see 2.2) have different control mechanisms and levels of detail internally. Upon closer examination of the implementation of the heat pump FMU generated for Dymola and the TRNSYS heat pump Types, this becomes clear. In contrast to the map-based Types described in section 2.1, which TRNSYS otherwise provides for heat pumps, the heat pump model inside Dymola describes the underlying dynamic thermodynamic processes.

The available TRNSYS heat pump Types can react immediately to changing simulation conditions and does not require any start-up phases. TRNSYS does not pay attention to internal thermodynamic processes and the heat pump is very flexible in terms of its application within the limits of the given map. The Modelica model, on the other hand, depicts the heat pump more realistically and reaches its limits in some modelled operating situations by TRNSYS. This can result in an error message and the cancellation of the simulation. The challenge here is to design the FMU for the more detailed heat pump model in such a way that it can deal robustly but still physically valid with the conditions in the TRNSYS model and, conversely, to limit the boundary conditions for the heat pump that arise during a simulation to an acceptable level for the FMU from the outset.

Other challenges are the differences in the user interface and in the correct coupling of the parameters, inputs and outputs of the heat pump FMU. The TRNSYS heat pump Types can generate temperature increases as well as thermal work and efficiencies with the help of input mass flows and temperatures and the stored map. With FMU's, on the other hand, there is the freedom to work with target temperatures from the outset and the heat pump reacts accordingly to comply with these target temperatures based on the given boundary conditions. The FMU can make the scaling of the heat pump output in partial operation and the operation of several identical heat pumps in parallel more realistic. This requires restructuring the currently used TRNSYS decks and adapting the existing control strategies.

Due to the integration of the physics-based heat pump FMU with its greater complexity the simulation times of the energy system model inside TRNSYS including the building model can increase. In the development process and when simulating with large timesteps inside TRNSYS it might be feasible to include the target physics-based heat pump model either as a static TRNSYS Type or as an interpolation map exported out of Dymola with the same FMU interface. Exemplary a map of the heat pump model built in Dymola was created and implemented in TRNSYS. Here, too, it became apparent that the TRNSYS deck still had to be adapted to the more realistic implementation of the partial load behavior. The available input parameters offer a higher degree of freedom. When configuring the heat pump in TRNSYS, care must be taken to avoid creating singular systems. The full advantages of the Dymola heat pump model, such as the possibility of internal fault detection and advanced control mechanisms, are lost in the process, which is why Co-Simulations is being pursued.

The different time constants within the building, the energy system and the heat pump require different solver timesteps. In TRNSYS the solver will use a fixed step size of 15 minutes, while the step size inside the FMU developed inside Dymola will be exported as a Co-Simulation FMU with an internal variable step solver like *DASSL* or *Cvode*. The timestep of the dynamic step size solvers will be much finer and adjusted accordingly based on the control of local errors to meet user specified tolerances [9], [10]. This ensures that the different dynamics within the stiff heat pump model can be resolved numerically efficiently while minimizing the needed steps to reduce calculation effort. These different step sizes enforce an integration error. The supply temperature of the heat pumps heat sink, for example will reach the setpoint after the heat pump has started and the control has adjusted the compressor speed. Therefore, the setpoint temperature may be reached 10 minutes after the start of the heat pump. If the temperature at the time of 15 minutes will be communicated to TRNSYS the integrated thermal energy will differ from the slow rising supply temperature within the heat pump. Therefore, a rolling mean temperature described in Section 3. is included. So that the supply temperatures of the heat pump after the 15-minute TRNSYS time step represents the energy transferred within this time step to ensure conservation of energy between the two models. This rolling mean temperature will be available as well as the energy integrals to the TRNSYS model. Future investigations will enable the estimation of the error due to different step sizes over the timeframe of one year. The biggest differences should be occurring if the heat pump will be switched on and off rapidly due to the higher dynamics.

Currently the fluid properties are calculated within the heat pump model with the *TILMedia Suite* for each fluid in the corresponding state. The mean fluid properties of the source and the sink will be communicated to TRNSYS for use in the energy system. These different states of fluid properties may lead to minor differences between the calculation results within the FMU and TRNSYS. The estimated errors for changes in the isobaric heat capacity and the density due to temperature changes are approximated with less than 1% of the energy content within the range of -20°C up to 80°C for water.

To analyze the performance of the heat pump during the TRNSYS simulation currently only the communicated performance metrics by the heat pump at the TRNSYS time steps are available. A detailed view of the underlying calculation steps in the FMU from within TRNSYS could not be accessed. As a solution the heat pump FMU could be simulated afterwards by passing the corresponding TRNSYS inputs to the FMU and recording further information's directly from within an FMU simulator. Alternatively, it may be possible to include an automate file logging of the FMU simulation results from within the TRNSYS Type while simulating in Co-Simulation mode for further investigations of the heat pump control strategy and performance.

The adaptation of both models to the required boundary conditions as well as the interfaces required with which both programs can work smoothly requires a high degree of exchange between the project partners and the iterative approximation of both program boundary conditions so that the advantages of both programs can be used within a simulation.

It is not only at the model level that there are challenges that are worth taking a closer look at. In the course of the TEN.efzn project, the *IfT* and *siz energieplus* are working on the creation of a higher-level simulation model with integrated heat pump FMU's. The competence for the detailed heat pump model with all thermodynamic processes lies with the IfT. The researchers of this institute hold the licenses for Dymola and for all necessary libraries, as well as the expertise. *Siz energieplus*, on the other hand, has licenses for TRNSYS and many years of experience with building simulations and district simulations. At the beginning of the project, it had to be ensured both institutions get a rough understanding of the other program, so that the key points could also be reflected in their own simulation program and the interface could be prepared. Thus, it was necessary not only to create an interface for data exchange at the program level, but also for the scientists behind it to gain a deeper understanding of the other program. The question is whether the solutions developed in the project can also be applied beyond the project and can be transferred to other projects. The dependencies of the licenses are still a hurdle for this.

4. Conclusion and outlook

The TEN.efzn research project aims to develop a novel, flexible heat pump module. To investigate the integration of the heat pump module into broader energy systems and operating strategies, as well as cascading, control and fault detection, various simulation tools will be coupled.

In terms of these research questions there is a need for a detailed heat pump model, which is based on actually available components of the refrigeration circuit and provides insights into the thermodynamic operating behavior under different load conditions. The Dymola program, based on the Modelica programming language, offers a suitable platform with a large number of integrated product libraries. However, the generation of different load states by means of a weather-data-coupled thermal building simulation in Dymola is challenging. TRNSYS, on the other hand, offers a good basis for generating load profiles of different building cubatures at different locations with its pre-processing program TRNBuild. Due to its modular structure, TRNSYS is also suitable for mapping entire energy systems with heating circuits, buffer storage tanks and renewable energies. Control logic and error detection as well as AI-supported error prevention are also to be considered during the heat pump design. In order to use the strengths of the individual simulation programs, they are to be coupled with each other across institutions.

The Functional Mock-up Interface for Dymola and TRNSYS, developed by the associated project partner TLK-Thermo GmbH, represents an approach for integration of a Co-Simulation FMU generated from a Dymola model into a TRNSYS model. The interface connects the two different calculation and solution approaches and enables the heat pump model in Dymola to be executed in a more time-resolved and thermodynamically detailed manner than the higher-level system of use cases in TRNSYS. Once at the end of each time step, input and output data such as mass flows, temperatures or control signals are exchanged between the two programs via FMU.

Nevertheless, challenges arise in coupling the two programs. For example, the conventional heat pump Types provided by TRNSYS, which calculate the efficiencies based on a map, react more flexibly to rapidly changing load conditions or load conditions outside their area of application. For the integrated heat pump FMU, on the other hand, some limit states are already terminating criteria of the simulation. The FMU heat pump also contains other necessary input values, which on the one hand enable more realistic control, but on the other hand cause difficulties in integrating them into existing TRNSYS models. In general, it can be said that both Dymola and TRNSYS must adapt to the other program characteristics so that the interface can run without errors in the course of Co-Simulation. A certain basic understanding and iterative exchange between the research institutions are essential for this.

In the future, a large overarching model with components of the different programs is to be created. This requires further work for the smooth coupling of the programs as well as the validation of possible errors that may arise from the different simulation methods.

Acknowledgements

The Projekt "TEN.efzn – Transformation of the Energy System in Lower Saxony" is funded by zukunft.niedersachsen, the joint science promotion program of the Lower Saxony Ministry of Science and Culture and the Volkswagen Foundation. The authors thank the ministry and project partners for funding and support. Responsibility is assumed by the authors for the content of this publication.

References

[1] Mara Magni, Fabian Ochs, Samuel de Vries, Alessandro Maccarini, and Ferdinand Sigg, "Detailed cross comparison of building energy simulation tools results using a reference office building as a case study," Eindhoven University of Technology, Nov. 2021. Accessed: Sep. 15, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378778821005442?via%3Dihub
[2] "TRNSYS 18 - Volume 1 - Getting Started," Solar Energy Laboratory, Univ. of Wsconsin-Madison; TRANSSOLAR Energietechnik GmbH; CSTV . Centre Scientifique et Technique du Batiment; TESS - Thermal Energy Systems Specialists, 2018.
[3] "What is TRNSYS?" [Online]. Available: https://www.trnsys.com/index.html
[4] "TRNSYS 18 - Volume 5 - Multizone Building Modeling with Type 56 and TRNBuild," Solar Energy Laboratory, Univ. of Wisconsin-Madison; TRANSSOLAR Energitechnik GmbH; CSTB - Centre Scientifique et Technique du Batiment; TESS - Thermal Energy System Specialists, LLC, 2023.
[5] Dentel, Arno, Kendel, Christina, Afjei, Dr. Thomas, and Wetter, Michael, "Trnsys User Manual TYPE 402," Transsolar Energietechnik GmbH, 2022.
[6] "TESSLibs 18 - Volume 08 - HVAC Library Mathematical References," TESS - Thermal Energy Systems Specialists, 2021.
[7] M. Dempsey, "Dymola for Multi-Engineering Modelling and Simulation," in *2006 IEEE Vehicle Power and Propulsion Conference*, Sep. 2006, pp. 1–6. doi: 10.1109/VPPC.2006.364294.
[8] "Modelica." Accessed: Nov. 12, 2025. [Online]. Available: https://modelica.org/
[9] L. Petzold, *A description of DASSL: A differential/algebraic system solver*. 1982.
[10] R. Serban and A. Hindmarsh, *Cvodes, the sensitivity-enabled ode solver in sundials*, vol. 6. 2005. doi: 10.1115/DETC2005-85597.
[11] Ingo Frohböse, "TIL Suite." Accessed: Nov. 12, 2025. [Online]. Available: https://www.tlk-thermo.com/en/software/til-suite
[12] M. Wetter, W. Zuo, T. S. Nouidui, and X. Pang, "Modelica Buildings library," *Journal of Building*

Performance Simulation, vol. 7, no. 4, pp. 253–270, Jul. 2014, doi: 10.1080/19401493.2013.765506.
[13] L. Maier *et al.*, “AixLib: an open-source Modelica library for compound building energy systems from component to district level with automated quality management,” *Journal of Building Performance Simulation*, vol. 17, no. 2, pp. 196–219, Mar. 2024, doi: 10.1080/19401493.2023.2250521.
[14] C. Nytsch-Geusen *et al.*, *Buildingsystems - Eine modular hierarchische Modell-Bibliothek zur energetischen Gebäude und Anlagensimulation*. 2016.
[15] F. Jorissen, G. Reynders, R. Baetens, D. Picard, D. Saelens, and L. Helsen, “Implementation and verification of the IDEAS building energy simulation library,” *Journal of Building Performance Simulation*, vol. 11, no. 6, pp. 669–688, Nov. 2018, doi: 10.1080/19401493.2018.1428361.
[16] *ibpsa/modelica-ibpsa*. (Nov. 06, 2025). Modelica. IBPSA. Accessed: Nov. 12, 2025. [Online]. Available: https://github.com/ibpsa/modelica-ibpsa
[17] Michael Wetter and Christoph van Treeck, *IEA EBC Annex 60: New Generation Computing Tools for Building and Community Energy Systems*. 2017. Accessed: Nov. 03, 2025. [Online]. Available: https://www.iea-annex60.org/pubs.html
[18] “Functional Mock-up Interface.” Accessed: Nov. 12, 2025. [Online]. Available: https://fmi-standard.org/
[19] “buildingSMART Technical,” buildingSMART Technical. Accessed: Nov. 12, 2025. [Online]. Available: https://technical.buildingsmart.org/
[20] C. Nytsch-Geusen *et al.*, “Template based code generation of Modelica building energy simulation models,” presented at the The 12th International Modelica Conference, Prague, Czech Republic, May 15-17, 2017, Jul. 2017, pp. 199–207. doi: 10.3384/ecp17132199.
[21] C. Nytsch-Geusen, J. Rädler, M. Thorade, and C. Ribas Tugores, “BIM2Modelica - An open source toolchain for generating and simulating thermal multi-zone building models by using structured data from BIM models,” presented at the The 13th International Modelica Conference, Regensburg, Germany, March 4–6, 2019, Feb. 2019, pp. 33–38. doi: 10.3384/ecp1915733.
[22] “IBPSA Project 1.” Accessed: Nov. 12, 2025. [Online]. Available: https://ibpsa.github.io/project1/
[23] Christian Vering, “Optimale Auslegung von Wärmepumpensysteme für Bestandsgebäude,” Institute for Energy Efficient Buildings and Indoor Climate, Aachen, Dissertation, 2023.
[24] Fabian Wüllhorst, Laura Maier, David Jansen, L. Kühn, Dominik Hering, and Dirk Müller, “BESMod - A Modelica Library providing Building Energy System Modules,” Institute for Energy Efficient Buildings and Indoor Climate; E.ON Energy Research Center; RWTH Aachen, Dallas, USA, PROCEEDINGS OF THE AMERICAN MODELICA CONFERENCE 2022, 28.10 2022.
[25] R. Kossel, “Hybride Simulation thermischer Systeme am Beispiel eines Reisebusses,” Technische Universität Braunschweig, Braunschweig, 2011.
[26] J. Eker *et al.*, “Taming heterogeneity - the Ptolemy approach,” *Proceedings of the IEEE*, vol. 91, no. 1, pp. 127–144, Jan. 2003, doi: 10.1109/JPROC.2002.805829.
[27] AVL List GmbH, “Model.CONNECT™.” [Online]. Available: https://www.avl.com/de-de/simulation-solutions/simulation-software/modelconnect
[28] “Functional Mock-up Interface Specification.” Accessed: Nov. 14, 2025. [Online]. Available: https://fmi-standard.org/docs/3.0.2/
[29] A. Elsheikh *et al.*, “Modelica-enabled Rapid Prototyping Via Trnsys,” presented at the 2017 Building Simulation Conference, Aug. 2013. doi: 10.26868/25222708.2013.1414.
[30] Nicolas Bernier, Bruno Marcotte, and Michaël Kummert, “Calling Python from TRNSYS with CFFI,” Polythechnique Montrea, May 2022. [Online]. Available: https://zenodo.org/records/6523104
[31] Edmund Widl and Wolfgang Müller, “Generic FMI-compliant Simulation Tool Coupling,” Center for Energy, AIT Austrian Institute of Technology, Austria; Institute of Analysis and Scientific Computing, TU Wien, Austria, 17.05 2017. [Online]. Available: https://2017.international.conference.modelica.org/proceedings/html/submissions/ecp17132321_WidlMuller.pdf
[32] Katharina Eder, Edmund Widl, Conrad Gaehler, Barbara Beigelboeck, and Florian Judex, “Co-Simulation for Control Design - A Case Study for Cross-Domain Collaboration,” AIT Austrian Institute of Technology, Vienna, Austria; Siemens Schweiz AG, Zug, Switzerland, Salt Lake City, 12.08 2016.
[33] Ingo Frohböse, “TILMedia Suite.” Accessed: Nov. 12, 2025. [Online]. Available: https://www.tlk-thermo.com/en/software/tilmedia-suite
[34] OGUNLEYE OLAOLUWA and UGWU ODIRACHUKWUNMA, “Successive Substitution Method,” Numerical Analysis, Clarkson University, NY, New York, Dec. 2022.