ALICE2Modelica - Automated Building Model Generation for Building Control and Simulation

Maximilian Mork*, Eziama Ubachukwu*, Jakob Benz*,
Philipp Althaus*, André Xhonneux*, Dirk Müller*,†

*Energy Systems Engineering (ICE-1), Forschungszentrum Jülich GmbH, Jülich, Germany
†Institute for Energy Efficient Buildings and Indoor Climate, RWTH Aachen University, Aachen, Germany
m.mork@fz-juelich.de, e.ubachukwu@fz-juelich.de, j.benz@fz-juelich.de,
p.althaus@fz-juelich.de, a.xhonneux@fz-juelich.de, dmueller@eonerc.rwth-aachen.de

***Abstract*—The fast and user-friendly generation of controller models is a crucial prerequisite for the widespread implementation and scalability of Model Predictive Control (MPC) for buildings. Apart from this, these models can be employed for evaluating other more and more relevant characteristics of buildings, for example, energy-saving potentials in user behavior or the building's flexibility potential to reduce or shift loads. In this work, the *ALICE2Modelica* toolchain is presented, which enables the automated and scalable generation of building models for use in Modelica-based building control and simulation. The toolchain is based on the developed mini-language *ALICE*, which constitutes a straightforward approach to describe geometrical information of the envelope of rooms. Based on *ALICE* files, floor plans in SVG format, and parametrizable Modelica room templates, the toolchain automatically generates Modelica room and building models including Heating, Ventilation and Air Conditioning (HVAC) systems. Therefore, it is suitable for the modeling of large-scale multi-zone buildings. In a case study, based on an existing parameter estimation module and data from building operation, selected parameters of the toolchain-generated Modelica models are calibrated and user behavior is evaluated for an office building with respect to energy efficiency and energy-saving potentials.**

***Index Terms*—building modeling, building simulation, building control, building MPC, Modelica**

I. Introduction

In 2020, buildings accounted for 36 % of the global energy consumption and 37 % of energy-related CO_2 emissions [1]. Heating, Ventilation and Air Conditioning (HVAC) systems contribute to a major extent to the building energy consumption and thus, offer a large potential for increasing energy efficiency in buildings [2]. As an energy-efficient control method, Model Predictive Control (MPC) has been implemented in buildings in the form of both simulative and practical applications, which have demonstrated the simultaneous improvement of thermal comfort and reduction of energy consumption, with energy savings ranging from approximately 15 to 50 % [3], [4]. The core part of MPC is a model of the building energy system to be controlled, based on which the MPC predicts future system states and calculates control inputs minimizing a cost function over a prediction horizon while integrating constraints and predicted disturbances. The main bottleneck for a widespread application of MPC in practice consists in the generation of the control-oriented building models [5], as the model generation part is the most challenging step in applying an MPC and accounts for the majority of the project time [6]. When transferring an MPC to another building, the components for the forecast, optimization and parameter/state estimation can be largely reused but the building models have to be generated once again in tailored modeling approaches due to the unique characteristics of every building. The practice-oriented, large-scale MPC project *Opti-Control-II*, which performed seven months of MPC in a Swiss office building, draws the conclusion that a "framework allowing the fast generation of MPC suitable models is a key factor to the widespread adoption of MPC in building control" [7].

Modelica [8] is a modeling language, which is capable of establishing a basis for generating control-oriented building models. It suits the use in MPC applications being an open-source, equation-based and object-oriented modeling language with a graphical interface for connecting components, which underlines its flexibility and user-friendliness. Nowadays, several building simulation libraries are available in Modelica, e.g., libraries based on the Modelica IBPSA library [9] such as the AixLib [10]. These libraries offer a wide range of reusable and shareable models for HVAC systems and buildings. Nevertheless, Modelica is not yet a common tool for companies that are specialized in implementing building automation.

A. Background

By employing control-oriented models, MPC approaches can be applied to buildings during the operating phase. Simulation models, which represent digital twins of the real buildings, allow for implementing MPC approaches as early as during the planning phase, to investigate different MPC configurations or to simulate energy demands to dimension building HVAC. Apart from this, simulation models allow for simulating the building behavior to evaluate user behavior with respect to energy-saving potentials or calculate the flexibility potential that buildings can provide by reducing or shifting their loads. Simulations can also be conducted for fault detection or within Hardware-in-the-Loop (HiL) approaches. Several authors and

This work was supported by the BMWK (Federal Ministry for Economic Affairs and Climate Action), Germany, promotional reference 03EGB0010A, and the European Union, FunSNM project, promotional reference 22DIT02.

research groups have developed tools for scalable modeling of building energy systems to be used in building control and simulation as listed in the following paragraphs.

In [11], a flexible model transformation system is implemented to link *Building Information Modeling* (BIM) with different Modelica libraries to support Building Energy Performance Simulation (BEPS). SimModel is used as BIM output format to convert the data into the XML-based format SimXML. Based on this format, mapping rule schemas are developed between SimModel and different Modelica libraries to handle the differences between the corresponding data models and generate simulation-ready Modelica models.

In [12], a toolchain is proposed, which uses BIM based on the *Industry Foundation Class* (IFC) format for the automated generation of Modelica grey-box structures. The grey-box model structures are formed according to a resistance-capacitance (RC) approach analogous to an electrical network. Starting from the simplest structure, different complexities in the form of additional resistances and capacitances are added until the model fulfills the validation criteria. During the model generation, the building's window and wall orientations as well as adjacent zones and systems are taken into account. BIM contains the characteristics of wall layers and material properties based on which the initial guesses for the optimization are calculated.

In [13], the toolchain *BIM2SIM* is developed to derive simulation-capable models from BIM models. Models are generated for the branches of thermal building simulation (both for EnergyPlus and Modelica), HVAC simulation (for Modelica), Computational Fluid Dynamics (CFD), and Life Cycle Assessment (LCA). For the simulation of the thermal building envelope, information is extracted from the *IFC4* format and enriched by user inputs based on templates. The output can be coupled with the HVAC simulation model, which is generated based on topological graph-based simplification.

In [6], the *Building Resistance-Capacitance Modeling* (BRCM) Matlab Toolbox is implemented for building modeling for use in MPC. The toolbox enables the fast generation of (bi-)linear resistance-capacitance models and separates the modeling of the buildings' thermal envelope from external heat fluxes. The thermal model is built based on building geometry and construction data, whereas the modeling of the external heat fluxes requires data from the building systems. The toolbox is limited to bilinear modeling, models building HVAC (e.g., floor heating or chilled ceilings) as simple controllable heat gains and radiative and convective heat transfer within a combined heat transfer coefficient.

In [14], a white-box Modelica modeling approach is proposed for use in building MPC, which can be refined using measurement data during operation. The modeling framework allows for nonlinear HVAC modeling and is based on a browser-based tool, which only allows valid building geometries to be entered. Building geometry information including orientations and surface areas is automatically derived from the export, while the user introduces additional information (e.g., material layers) via the graphical user interface.

The challenge to automatically generate simulation-capable models based on standardized input formats also applies to other domains. For example for power grid systems, frameworks to automatically create Modelica models based on the standardized Common Information Model (CIM) have been developed in [15]–[17].

B. Contribution

The workflow for the *ALICE2Modelica*[1] toolchain begins with capturing the geometrical and internal component information of building spaces using a specially designed mini-language called *ALICE* [18]. *ALICE* enables quick and scalable description of these spaces by nonspecialist human operators through supporting an intuitive specification of dimensions and positions of components like radiators, windows, and doors. This approach is particularly scalable for office buildings with a regular pattern. A JSON-formatted serialization of the geometrical and component information of the building spaces is then exported by *ALICE* through a web-based API.

In a second step, based on the *ALICE* output, the proposed Python-based toolchain *ALICE2Modelica* enables an automated and scalable generation of Modelica room and building models for simulation and optimization. The toolchain builds on Modelica template models for the HVAC and material layers, which can be parameterized via Python by actuator- and building-specific data sheets. Compared to manual modeling approaches, the modeling time and the risk of modeling errors (e.g., unconnected or incorrectly connected Modelica ports, incorrect specification of orientations) are reduced. The toolchain is applicable to existing and new buildings in contrast to many of the aforementioned modeling approaches focusing on new buildings. It builds upon a straightforward and user-friendly mini-language and does not require the existence of BIM data, which may not be available at all or not complete for (older) buildings. The output of the toolchain are nonlinear white-box models, formulated based on the Modelica library AixLib [10]. To be able to calibrate the model, an interface to an existing parameter estimation module is implemented, which calibrates selected model parameters based on measurements from building operation on the campus of Forschungszentrum Jülich (FZJ). In a final case study, the calibrated toolchain output models are employed for evaluating user behavior with respect to energy efficiency based on measured occupant data.

II. Methodology

A. ALICE

ALICE is a human-oriented mini-language as well as a web-based tool for generating geometrical diagrams of building spaces and their internal components (see Fig. 1). The primary motivation for the development of the tool was to allow non-specialist users to easily and quickly capture the relevant

[1]The open-source toolchain including exemplary ALICE and SVG input files can be found here: https://jugit.fz-juelich.de/iek-10/public/tools/alice2modelica

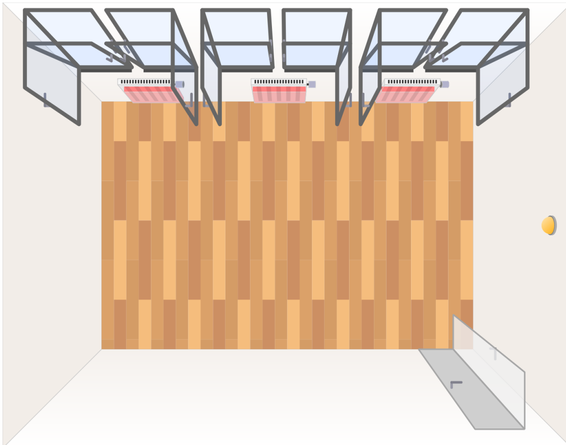

Fig. 1. A pseudo-3D diagram of a room generated by *ALICE*.

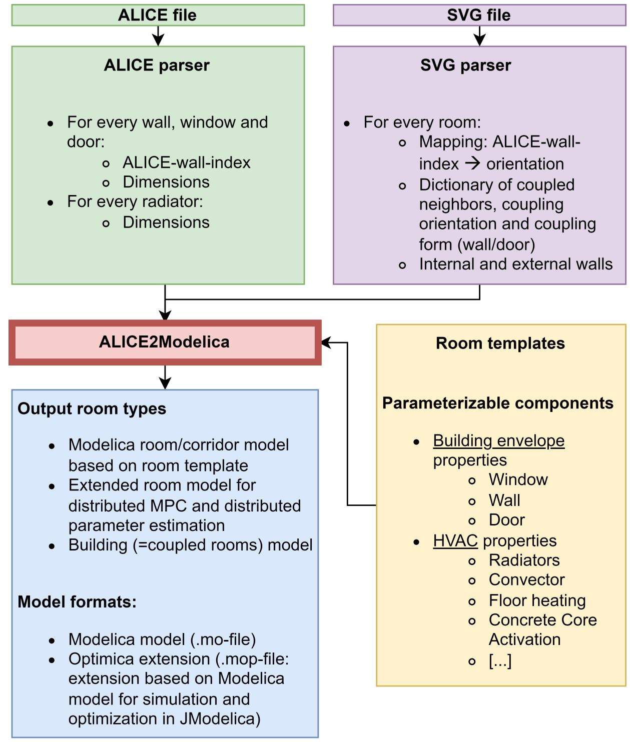

Fig. 2. Overview of *ALICE2Modelica* toolchain

geometry of the building space for visualization, as well as the positions and types of the internal components since the required details about the internal structure of the rooms at FZJ are not available in any (accessible or comprehensive) database. Additionally, *ALICE* automatically associates the room components (e.g., radiators and windows) with their corresponding sensor and actuator devices by leveraging the structured naming scheme for the devices. For the FZJ campus, the device information is retrieved from an internal device management tool called WALDO [19].

In *ALICE*, positions of components are described by the human operator relative to other structures in the room, e.g., by saying "the radiator is centered 20 cm below windows 1 and 2" in the equivalent *ALICE* syntax, or else as absolute positions using the coordinate system of the "host surface". Host surfaces are walls, floor, and ceiling, while the *airvolume* serves as a special virtual host. The hierarchical structure of the *ALICE* representation defines these top-level hosts as direct children of the space object; all other components are "hosted" by these surfaces and the air volume. In *ALICE*, each surface is treated separately, having its own 2D coordinate system where the lower left corner is the origin when facing the surface. *ALICE* currently only captures rectangular spaces. The wall opposite the main door leading into the space is treated as *wall 1*, and then increments clockwise to *wall 4* (left wall when standing at the main entrance). The floor and ceiling are surfaces 0 and 5 respectively.

The generated room component and geometry information, as well as the sensor IDs for each component, are exported by *ALICE* in an own JSON format via a web-based API, where all the relative positioning information captured in the language is translated into absolute coordinates using the local coordinate system of the host surface.

B. *ALICE2Modelica*

This section details the different inputs, components and features of the *ALICE2Modelica* toolchain. An overview of the Python-based toolchain including input and output formats is given in Fig. 2.

The first input to the toolchain is an individual *ALICE* file for every room (upper green box in Fig. 2). An *ALICE* parser extracts the relevant information from the file for every room: For every wall, window and door, the ALICE-specific wall index is extracted determining the wall affiliation of each component. Additionally, the dimensions of these components and potential radiators are obtained.

The second input is an SVG file for every floor of the building including window and door sensors (upper purple box in Fig. 2). An SVG parser extracts the following information for every room: For every *ALICE* wall index of every room, the corresponding orientation is determined. An additional dictionary is generated, which, for every room, contains the coupled neighbor rooms, respective coupling orientation and coupling form. The latter refers to the thermal coupling via a door or wall. Finally, the SVG parser derives the information about which room walls are external and internal.

The third input to the toolchain are Modelica room templates (yellow box in Fig. 2), which include parameterizable models for material properties (e.g., walls or windows) and HVAC systems (e.g., radiators or floor heating). These room templates form the base for the automated model generation, are based on the AixLib library and may be parameterized in Python based on standardized data sheets in Modelica. In the current implementation, the template models build upon the Modelica library AixLib, but the basic functionality could also be transferred to other Modelica libraries (such as [20]–[22]) using the corresponding Modelica component equivalents in the respective library. Beyond that, the information extracted from the ALICE and SVG files (see Fig. 2) are stored in a generalized format which would also allow for creating other output model formats than Modelica by mapping the Modelica components onto the equivalents of the respective modeling language.

Within the *ALICE2Modelica* toolchain, the following outputs are generated (blue box in Fig. 2):

- Room simulation models
- Room optimization models (for decentralized and distributed MPC and parameter estimation)
- Building (=coupled rooms) simulation model
- Building (=coupled rooms) optimization model (for centralized MPC and parameter estimation)

First, the Modelica simulation models for the individual rooms (and corridors) are created by originating from the room templates. Based on the orientation and dimension information extracted from the *ALICE* and SVG files, the corresponding external/internal walls and windows are added to the room. For every window facade of the room, control inputs for solar shading are added to enable active control of solar blinds according to the models developed in [23]. The room models are generally equipped with half inner walls to facilitate the coupling of rooms in an overall coupled rooms model.

Based on the created room simulation models in the first step, the optimization models for a local, decentralized room MPC are created by making use of the *Optimica* .mop-file extension, which allows for simulation and optimization in the JModelica.org framework [2] [25]. The room model generation is completed with a compilation check of the created instances for the simulation and optimization models in JModelica.org, based on the Functional Mock-up Unit (FMU) format [26] for simulation and the CasADi-based optimization format.

Next, for use in a distributed MPC and parameter estimation, local controller models are created (second output room type in the blue box in Fig. 2) based on the generated room models in the first step. The controller models are generated to be compatible with the distributed MPC approach that has been developed in [27], is based on JModelica.org and uses Alternating Direction Method of Multipliers (ADMM) and Nash Equilibrium but the controller models also suit general distributed room-based MPC approaches. A distributed MPC is implemented to reduce computation time compared to a centralized variant and enhance scalability, adaptability and reliability for large-scale buildings. For every thermally coupled neighbor room, the second half of the inner wall is added and connected in addition to a Modelica *PrescribeTemperature* model for prescribing the temperatures of the corresponding neighbor rooms. If the information extracted from the SVG parser specifies that the corresponding rooms are also coupled via a door, a data-driven door model according to [27] is added. The controller models are created both in the Modelica and JModelica-specific *Optimica* format. In analogy with the simulation models created in the first step, a compilation check of the FMU and the CasADi-based transferred optimization problem is performed.

In the final step, the overall Modelica building model consisting of all coupled rooms is created (third output room type in the blue box in Fig. 2). Looping over all rooms extracted from ALICE, each room is added to the global building model and coupled to its neighbors. According to the coupling between the rooms extracted by the SVG-parser, the room models are thermally coupled via connecting their half-wall ports or, in case of door coupling, adding a door model. The step is completed by a compilation check of the created FMU instance and the transferred optimization problem.

III. Case Study

A. Calibration

In a case study, the ALICE2Modelica toolchain is executed for an office building on the campus of Forschungszentrum Jülich. According to Fig. 2, for every room, the number and dimensions of all walls, windows and doors are extracted from the room-individual ALICE files. Based on the SVG files for the different floors of the building, the orientations of all walls and the coupling between the individual rooms are specified including the characteristic of each wall being an internal or external wall. The office building is part of a set of buildings on the campus that have been equipped with additional sensors and actuators in the context of the Living Lab Energy Campus (LLEC) project [28]. In the following step, the generated room models are interfaced with an existing nonlinear Modelica-based calibration module [3], where a user-defined subset of model parameters is estimated based on measurement data from building operations. In the calibration procedure presented in [3], each parameter is calculated in a specific period, where it has a large influence on the room dynamics and the impact of other parameters is reduced. In contrast to this approach, a more straightforward procedure is pursued here, where, for each parameter, the final value is determined by averaging the respective estimated values from all executed estimations over the evaluation horizon.

The rooms of the office building are equipped with radiators for heating and Venetian blinds for solar shading. As input measurement data, weather data from a weather station of a nearby building [3], room temperatures and CO_2 concentration, heating supply temperatures as well as the opening states of windows and doors m are used. For the window openings, an average room-specific value is calculated, which considers the number of windows and the opening angle (closed/tilted/fully open). In addition, the measured control inputs for heating in the form of the valve opening of the radiators and for shading in the form of the vertical shading position and slat inclination angle are integrated. At last, measurements for CO_2 concentration in the rooms and corridor, opening states of windows and doors as well as the room volumes are used to calculate the historic number of occupants based on a CO_2 balance according to [29]. Input to the occupants' calculation is an experimentally validated set of parameters for air exchange via window opening and door opening.

[2] The JModelica.org framework is discontinued by Modelon. The continuation OPTIMICA Compiler Toolkit can be found in [24]

The rooms are calibrated locally in a distributed calibration scheme. In the local room parameter estimations, thermal coupling with neighboring rooms is considered by integrating the thermally coupling inner walls and doors as well as the room temperatures of the respective neighboring rooms. After estimating all parameters, the rooms exchange the individually calculated values for the thermal conductivity of the inner walls. These parameters constitute a shared parameter, which should have an identical value for all local calibrations to maintain physical consistency. Based on the forwarded room-calculated values, an average value is calculated, which is set is set fixed in the local estimations and the remaining parameters are recalculated in a final step. Overall, the following room parameters are estimated:

- Infiltration air exchange
- Ventilation air exchange for fully opened windows
- Nominal power of the radiator
- Coefficients for thermal (U-value) and solar energy transmission (g-value) for windows
- Thermal conductivity for external and internal walls (one-layer configuration respectively)

The calibration is executed for half of a wing of a floor of the office building consisting of 10 adjacent offices in southwest orientation. For a prediction horizon of 24 h of each parameter estimation, the calibrated room models exhibit an Root Mean Square Error (RMSE) of 0.77 °C for the final estimated parameter set (averaged for all rooms and over the considered period from April 11 to May 4, 2024.). This period comprises ambient temperatures from -1 to 28 °C and thereby, covers a wide range of operating points. According to [30], an RMSE for reproducing the building envelope below 1 °C indicates a sufficient model accuracy for use in building MPC.

B. *Evaluation of user behavior*

In a subsequent step, the calibrated room models are employed for evaluating user behavior with respect to energy efficiency. This study is meant to reveal energy-wasting schemes in user behavior and allow to sensitize the building occupants with specific insights. The analysis focuses on the occupant's heating and ventilation behavior and accesses the calibration input data and especially, binary occupancy, manually operated window openings and the room temperature heating set-points, which can be specified by the occupants. The energy consumption of the calibrated Modelica models is simulated once based on the measured occupant data and once with assumed ideal, energy-efficient user inputs for comparison.

For the simulated heating control and both the real and ideal occupant behavior, a PI-controller controls the radiator valve opening by tracking the specified room temperature set-points. For the ideal heating behavior, time-variant room temperature set-points are specified. In times of occupancy (and a lead time of 1.25 h to account for the inertia of the radiator heating to provide the requested room temperature), the set-point is set to 20 °C, being the minimum temperature for workplaces for physically easy and mainly seated activity according to the technical rules for workplaces of the German Federal Institute for Occupational Safety and Health [31]. For times without occupancy, 17 °C is set. These are also the thresholds (without the heating lead time), below which thermal discomfort is assumed for the occupants.

For the simulation of the real ventilation behavior, the measured user interactions with the windows are included to calculate the thermal exchange with the the environment based on the previously calibrated ventilation air exchange for fully opened windows. For the ideal ventilation behavior, a Rule-Based Control (RBC) with hysteresis is implemented, which fully opens half of the windows when the room CO_2 concentration exceeds a value of 1 000 ppm [3] and closes the windows when falling below a value of 800 ppm. For air quality, values above 1 000 ppm are assumed to cause air quality discomfort.

For an office in southwest orientation that has been selected due to energy-inefficient user behavior, the simulation results for April 17, 2024 are depicted in Fig. 3 for both the real and ideal occupant behavior. For the accounted simulation horizon, the real occupant consumes 0.74 kWh more heating energy than the ideal occupant, as the room is heated throughout the day based on the high room temperature set-point. At the same time, CO_2 discomfort is reduced for the ideal occupant by 16 016 ppmh to 1 578 ppmh by fully opening half of the windows exceeding a threshold of 1 000 ppm. Accumulated thermal discomfort is zero for both behavior profiles.

In the ”Room temperature” subplot, it is shown that the real temperature set-points are notably higher than the ideal ones and in addition, there is no automatic set-back during night or non-occupancy for the real user inputs. In the future, the room-set points could be adjusted to presence schedules filled by the occupants in the web-based interface *JuControl* [32] to avoid wasting energy during the absence of the occupants. The ”Radiator valve opening” subplot exhibits the valve opening angle, which is controlled by the PI-controller to track the respective room temperature set-points. The subplots ”CO_2 concentration” and ”Window opening” describe the user ventilation behavior (based on a room-averaged window opening) and show that via the ideal ventilation behavior, the CO_2 discomfort is significantly reduced and at the same time, energy losses during non-occupancy are reduced.

In Table I the user behavior evaluation results are exhibited for all calibrated rooms for a period of 19 days from April 16 to May 4, 2024. The table includes information on deviations in average temperature set-points, window opening and differences in thermal discomfort, CO_2 discomfort and heating energy consumption.[4] The difference quantities respectively calculate the difference between the real and ideal occupant, thus, a positive value in the last three columns indicates an improvement (both for discomfort and energy consumption).

[3] Pettenkofer threshold

[4] The heating power is likely underestimated as an effect of the possibly underestimated measurements of the heating supply temperature. Nevertheless, these temperatures were also input for the calibration and relative comparisons are unaffected.

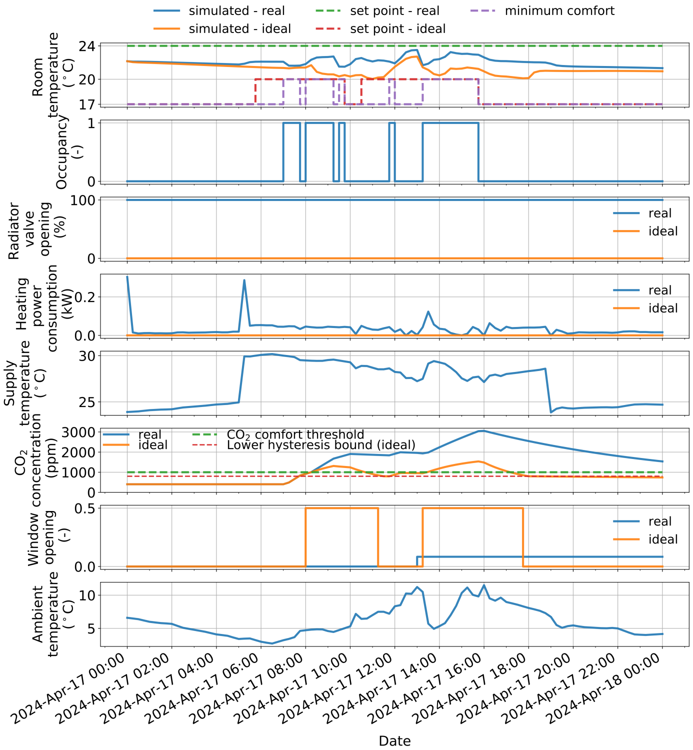

Fig. 3. Comparison of real and ideal user behavior for a single office

TABLE I
USER BEHAVIOR ANALYSIS BY EVALUATING SAVED HEATING ENERGY CONSUMPTION AS A FUNCTION OF DEVIATIONS IN TEMPERATURE SET-POINTS, WINDOW OPENING, THERMAL DISCOMFORT AND CO_2 DISCOMFORT (Δ BETWEEN REAL AND IDEAL BEHAVIOR IN THIS ORDER)

Room	Δ ∅temp. set-point (K)	Δ ∅wind. opening (-)	Δ therm. discomf. (Kh)	Δ CO_2 discomf. (ppmh)	Δ heating ener. cons. (kWh)/ rel. change (%)
Office 1	2.3	-0.05	-0.08	250*10E3	3.1/ 51 %
Office 2	6.7	-0.02	0	379*10E3	8.2/ 99 %
Office 3	4.8	-0.03	0	173*10E3	4.8/ 100 %
Office 4	4.8	-0.04	0	302*10E3	8.9/ 100 %
Office 5	3.6	-0.07	0	707*10E3	3.6/ 96 %
Office 6	1.5	-0.05	0.02	201*10E3	-1.9/ -191 %
Office 7	4.8	-0.01	0	87*10E3	2.4/ 100 %
Office 8	3.6	-0.07	0	632*10E3	1.4/ 100 %
Office 9	2.8	0.08	-2.14	85*10E3	4.2/ 78 %
Office 10	4.7	0.02	-8.99	55*10E3	10.7/ 73 %

The results exhibited in the table indicate a strong correlation between the deviations in temperature set-points and heating energy consumption. For office 2 with a set-point deviation of 6.7 K, office 3 with 4.8 K, office 4 with 4.8 K and office 10 with 4.7 K, the highest savings can be achieved with 8.2, 4.8, 8.9 and 10.7 kWh. The deviations in the window openings are mostly negative, since, similar to the behavior in Fig. 3, the ideal occupant ventilates more than the real one to reduce CO_2 discomfort. In office 6 with a rather small temperature set-point deviation, the increased ventilation rates of the ideal occupant lead to higher heating energy consumption (with improved CO_2 comfort). The increased thermal discomfort in offices 9 and 10 results from increased ventilation rates during occupancy; however, the discomfort amount is small with respect to the evaluation horizon.

IV. Conclusion and Outlook

This paper presents *ALICE2Modelica*, a toolchain for automated and scalable generation of Modelica buildings models for use in control and simulation. The toolchain is based on the developed mini-language *ALICE*, which allows for efficient and user-friendly description of geometrical information of rooms. As output, the toolchain generates nonlinear white-box Modelica models of the individual rooms, local controller models for distributed MPC or parameter estimation and a building model composed of all coupled rooms. The output can be used both for simulation and (distributed or centralized) optimization. For simulation, FMUs are created, and for optimization, JModelica-compatible optimization formats are deployed. Apart from the *ALICE* files, inputs for the toolchain are SVG files of the building and Modelica room templates, where building-specific HVAC and material properties can be specified. In a final case study, the toolchain is executed for an office building on campus and interfaced with an existing calibration module to estimate selected parameters of the output Modelica models. In a simulation study, the calibrated Modelica models are employed for evaluating energy-saving potentials by comparing real and ideal, energy-efficient user behavior. Referring to potential extensions and use cases of the toolchain, the output of ALICE2Modelica and the calibration module could be used as a basis for (real-world) distributed MPC as presented in [3], [27].

References

[1] United Nations Environment Programme, "2021 Global Status Report for Buildings and Construction: Towards a Zero-emission, Efficient and Resilient Buildings and Construction Sector," Nairobi, 2021. [Online]. Available: https://globalabc.org/resources/publications/2021-global-status-report-buildings-and-construction

[2] M. W. Ahmad, M. Mourshed, B. Yuce, and Y. Rezgui, "Computational intelligence techniques for HVAC systems: A review," *Build. Simul.*, vol. 9, pp. 359–398, 2016.

[3] M. Mork, F. Redder, A. Xhonneux, and D. Müller, "Real-world implementation and evaluation of a Model Predictive Control framework in an office space," *J. Build. Eng.*, vol. 78, p. 107619, 2023.

[4] J. Drgoňa, J. Arroyo, I. Cupeiro Figueroa, D. Blum, K. Arendt, D. Kim, E. P. Ollé, J. Oravec, M. Wetter, D. L. Vrabie, and L. Helsen, "All you need to know about model predictive control for buildings," *Annu. Rev. Control*, vol. 50, pp. 190–232, 2020.

[5] E. Atam and L. Helsen, "Control-Oriented Thermal Modeling of Multizone Buildings: Methods and Issues: Intelligent Control of a Building System," *IEEE Control Syst.*, vol. 36, no. 3, pp. 86–111, 2016.

[6] D. Sturzenegger, D. Gyalistras, V. Semeraro, M. Morari, and R. S. Smith, "BRCM Matlab Toolbox: Model generation for model predictive building control," in *2014 Am. Control Conf.* IEEE, 2014, pp. 1063–1069.

[7] D. Sturzenegger, D. Gyalistras, M. Morari, and R. S. Smith, "Model Predictive Climate Control of a Swiss Office Building: Implementation, Results, and Cost-Benefit Analysis," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 1, pp. 1–12, 2016.

[8] S. E. Mattsson and H. Elmqvist, "Modelica - An International Effort to Design the Next Generation Modeling Language," *IFAC Proc. Vol.*, vol. 30, no. 4, pp. 151–155, 1997.

[9] M. Wetter, C. van Treeck, L. Helsen, A. Maccarini, D. Saelens, D. Robinson, and G. Schweiger, "IBPSA Project 1: BIM/GIS and Modelica framework for building and community energy system design and operation – ongoing developments, lessons learned and challenges," *IOP Conf. Ser. Earth Environ. Sci.*, vol. 323, no. 1, p. 012114, 2019.

[10] L. Maier, D. Jansen, F. Wüllhorst, M. Kremer, A. Kümpel, T. Blacha, and D. Müller, "AixLib: an open-source Modelica library for compound building energy systems from component to district level with automated quality management," *J. Build. Perform. Simul.*, vol. 0, no. 0, pp. 1–24, 2023.

[11] J. Cao, R. Wimmer, M. Thorade, T. Maile, J. O'Donnell, J. Rädler, J. Frisch, and C. Van Treeck, "A flexible model transformation to link bim with different modelica libraries for building energy performance simulation," in *14th Int. Conf. IBPSA - Build. Simul. 2015, BS 2015, Conf. Proc.*, 2015, pp. 434–441.

[12] A. Andriamamonjy, R. Klein, and D. Saelens, "Automated grey box model implementation using BIM and Modelica," *Energy Build.*, vol. 188-189, pp. 209–225, 2019.

[13] D. Müller, D. Jansen, C. van Treeck, E. Fichter, V. Richter, B. Lüdemann, P. Jahangiri, J. Brunkhorst, M. Dahncke, and C. Warnecke, "BIM2SIM - Methodenentwicklung zur Erstellung von Simulationsmodellen aus Daten des Building Information Modeling," Tech. Rep., 2021.

[14] F. Jorissen, D. Picard, K. Six, and L. Helsen, "Detailed White-Box Non-Linear Model Predictive Control for Scalable Building HVAC Control," in *Proc. 14th Model. Conf. 2021, Linköping, Sweden, Sept. 20-24, 2021*, 2021, pp. 315–323.

[15] F. J. Gómez, L. Vanfretti, and S. H. Olsen, "CIM-Compliant Power System Dynamic Model-to-Model Transformation and Modelica Simulation," *IEEE Trans. Ind. Informatics*, vol. 14, no. 9, pp. 3989–3996, sep 2018.

[16] F. J. Gómez, L. Vanfretti, M. Aguilera, and S. H. Olsen, "CIM-2-mod: A CIM to modelica mapping and model-2-model transformation engine," *SoftwareX*, vol. 9, pp. 161–167, jan 2019.

[17] L. Razik, J. Dinkelbach, M. Mirz, and A. Monti, "CIMverter–a template-based flexibly extensible open-source converter from CIM to Modelica," *Energy Informatics*, vol. 1, no. S1, p. 47, oct 2018.

[18] E. Ubachukwu, J. Pick, A. Xhonneux, and D. Müller, "ALICE: A human-centric mini-language and tool for rapid representation and visualization of building spaces," *unpublished*, 2024.

[19] F. Redder, P. Althaus, E. Ubachukwu, S. Johnen, M. Mork, C. Küpper, P. Lieberenz, M. Oden, L. Westphal, T. Storek, A. Xhonneux, and D. Müller, "Information and Communication Technologies (ICT) for Holistic Building Energy System Operation in Living Labs: Conceptualization, Implementation, and Evaluation," 2024. [Online]. Available: http://dx.doi.org/10.2139/ssrn.4743282

[20] F. Jorissen, G. Reynders, R. Baetens, D. Picard, D. Saelens, and L. Helsen, "Implementation and verification of the IDEAS building energy simulation library," *J. Build. Perform. Simul.*, vol. 11, no. 6, pp. 669–688, 2018.

[21] M. Wetter, W. Zuo, T. S. Nouidui, and X. Pang, "Modelica Buildings library," *J. Build. Perform. Simul.*, vol. 7, no. 4, pp. 253–270, 2014.

[22] C. Nytsch-Geusen, J. Huber, M. Ljubijankic, and J. Radler, "Modelica BuildingSystems - eine Modellbibliothek zur Simulation komplexer energietechnischer Gebudesysteme," *Bauphysik*, vol. 35, no. 1, 2013.

[23] M. Mork, A. Xhonneux, and D. Müller, "Hierarchical Model Predictive Control for complex building energy systems," *Bauphysik*, vol. 42, no. 6, pp. 306–314, 2020.

[24] Modelon, "OPTIMICA Compiler Toolkit." [Online]. Available: https://help.modelon.com/latest/reference/oct/

[25] J. Åkesson, K. E. Årzén, M. Gäfvert, T. Bergdahl, and H. Tummescheit, "Modeling and optimization with Optimica and JModelica.org-Languages and tools for solving large-scale dynamic optimization problems," *Comput. Chem. Eng.*, vol. 34, no. 11, pp. 1737–1749, 2010.

[26] C. Andersson, J. Akesson, and C. Fuhrer, "PyFMI: A Python Package for Simulation of Coupled Dynamic Models with the Functional Mock-up Interface," *Tech. Rep. Math. Sci.*, vol. 2016, no. 2, pp. 1–40, 2016.

[27] M. Mork, A. Xhonneux, and D. Müller, "Nonlinear Distributed Model Predictive Control for multi-zone building energy systems," *Energy Build.*, vol. 264, p. 112066, 2022.

[28] P. Althaus, F. Redder, E. Ubachukwu, M. Mork, A. Xhonneux, and D. Müller, "Enhancing Building Monitoring and Control for District Energy Systems: Technology Selection and Installation within the Living Lab Energy Campus," *Appl. Sci.*, vol. 12, no. 7, 2022.

[29] D. Calì, P. Matthes, K. Huchtemann, R. Streblow, and D. Müller, "CO_2 based occupancy detection algorithm: Experimental analysis and validation for office and residential buildings," *Build. Environ.*, vol. 86, pp. 39–49, 2015.

[30] D. Blum, Z. Wang, C. Weyandt, D. Kim, M. Wetter, T. Hong, and M. A. Piette, "Field demonstration and implementation analysis of model predictive control in an office HVAC system," *Appl. Energy*, vol. 318, p. 119104, 2022.

[31] Federal Institute for Occupational Safety and Health, "Technical rules for workplaces - Room temperature (ASR A3.5)," 2022. [Online]. Available: https://www.baua.de/DE/Angebote/Regelwerk/ASR/ASR-A3-5.html

[32] E. Ubachukwu, J. Pick, L. Riebesel, P. Lieberenz, P. Althaus, D. Müller, and A. Xhonneux, "LLEC Energy Dashboard Suite: User Engagement for Energy-Efficient Behavior using Dashboards and Gamification," *36th Int. Conf. Effic. Cost, Optim. Simul. Environ. Impact Energy Syst. ECOS 2023*, no. June, pp. 3241–3252, 2023.