Usefulness of the obFMU Module Examined through a Review of Occupant Modelling Functionality in Building Performance Simulation Programs

Andrew Cowie[1], Tianzhen Hong[2], Xiaohang Feng[3], Quentin Darakdjian[4]
[1]University of Strathclyde, Glasgow, UK
[2]Lawrence Berkeley National Laboratory, Berkeley CA, USA
[3]Tsinghua University, Beijing, China
[4]AI Environnement, Val de Fontenay, France and University of La Rochelle, LaSIE UMR CNRS 7356, France

Abstract

Whilst deterministic models such as prescribed schedules and black-box controls, are the traditional methods for representing occupant behavior in building performance simulation (BPS), recent trends have seen more stochastic models emerging. However implementation of these models in BPS software has been sporadic. This paper presents a review of occupant behavior modelling capabilities of popular BPS programs to clarify this issue, and uses these results to examine the usefulness of the obFMU module, recently released under the remit of IEA-EBC Annex 66.

Introduction

A wide array of BPS programs are available today to simulate energy and environmental performance of buildings. The building energy software tools (BEST) directory (2016) lists 45 whole-building energy simulation programs. There are both commonalities and differences in the modelling capabilities to support performance evaluation in the building life cycle (e.g. Crawley et al. 2008; Zhou et al. 2014; Zhu et al. 2013). The basic features to represent occupant related inputs in a deterministic or static way are generally fairly consistent. For example, each space typically has input of number of occupants and whole-year occupant schedules, as well as schedules of space air temperature and humidity setpoints, or some equivalent prescriptive functionality. Lighting and plug-load equipment are generally represented with design or peak power and/or schedules. Operation of windows, shading devices and other operable systems are usually described by deterministic rules based on indoor and outdoor environmental parameters and/or schedules.

Aside from these prescriptive techniques, a number of researchers such as Hong et al. (2016a) and Gilani et al. (2016) have recognized that the uptake and implementation of stochastic occupant modelling in these programs has generally been slow and inconsistent, leading to strongly heterogeneous functionality in this respect among BPS programs. Many lack such capability, some use simple random number generators and some have built-in comprehensive probabilistic occupant behavior models, whilst some enable co-simulation with dedicated occupant behavior modelling tools. As noted by Hong et al. (2016b) such models can help to bridge the gap between simulation predictions and reality; Gilani et al. (2016) concluded from a comparison of stochastic and static occupant models in building simulation that dynamic models are imperative for simulation-supported design and code compliance. Jang and Kang (2016) found that use of stochastic modelling can reduce discrepancies in building energy prediction, Virote and Neves-Silva (2012) found that dynamic models provided better predictions of energy use than static models, and Hoes et al. (2009) concluded that more accurate characterization of occupant behavior in simulation can result in a more optimal building design. Furthermore, Yan et al. (2015) have highlighted the need for standardized input structures and functionality to improve potential for model exchange or reuse.

As part of IEA-EBC Annex 66, Hong et al. (2016c) have recently released a computational tool called occupant behavior functional mock-up unit (obFMU). An FMU is defined by the functional mock-up interface (FMI) standard (FMI, 2016); this is a “tool independent standard to support both model exchange and co-simulation of dynamic models”. Hence, obFMU essentially seeks to provide a platform for implementing occupant behavior models (deterministic or stochastic) that can co-simulate in a consistent manner with any BPS program that has implemented the FMI standard. This has potential to address the need for consistent and standardized occupant behavior modelling functionality in BPS programs.

This paper presents a review of occupant behavior modelling capabilities in a selection of current BPS programs. The perceived gaps in functionality are clarified, and specific limitations are discussed. Whilst in this paper the review is used to examine the usefulness of obFMU in particular, they may also be useful to inform other research.

https://doi.org/10.26868/25222708.2017.504

Table 1: Summary of occupant movement and/or presence modelling functionality.

Program	Stochastic functionality	Deterministic functionality
DeST	Markov chain occupant movement/presence model (Wang et al., 2011; Feng et al. 2015).	Prescribed schedules of number of occupants.
DOE-2.1E	Can implement user-defined models via external function interaction.	Prescribed schedules of number of occupants.
EnergyPlus (U.S. Department of Energy, 2016)	Can implement user-defined models via: 1) (open) source code modification, 2) proprietary model input language (EMS), and 3) standardised co-simulation interfaces (FMI, BCVTB).	Prescribed schedules of number of occupants.
ESP-r (Hand, 2015).	Gaussian distribution of arrival and departure times via bespoke coupling with external code (SHOCC; Bourgeois, 2005). Can implement user-defined models via (open) source code modification.	Prescribed schedules of occupant casual gains. Input from monitoring sensors (Clarke et al., 2014).
IDA-ICE	Can implement user-defined models via standardised model input language (NMF).	Occupancy can be characterised as a control system, e.g. input from monitoring sensors. (EQUA Simulation AB, 2016). Prescribed schedules.
IES-VE	Can implement user-defined models via proprietary model input language (APpro).	Prescribed schedules of percentage or number of occupants.
Pleiades + Comfie (IZUBA énergies, 2016)	None.	Prescribed schedules of number of occupants.
TRNSYS (TRNSYS17, 2012)	Can implement user-defined models via: 1) source code modification, 2) external function interaction (e.g. Baeten and Saelens, 2011), 3) proprietary model input language (W; Keilholz et al., 2009).	Prescribed schedules of number of occupants.

Table 2: Summary of lighting use modelling functionality.

Program	Stochastic functionality	Deterministic functionality
DeST	Operation probability related to event or environment (Wang, 2014; Ren et al., 2014; Wang et al., 2015).	Prescribed schedules of light operation.
DOE-2.1E	Can implement user-defined models via external function interaction.	Prescribed schedules of light operation. Control based on required workplane illuminance.
EnergyPlus (U.S. Department of Energy, 2016)	Can specify schedules of operation probabilities. Can implement user-defined models via: 1) (open) source code modification, 2) proprietary model input language (EMS), and 3) standardised co-simulation interfaces (FMI, BCVTB).	Prescribed schedules of light operation. Control based on required workplane illuminance.

ESP-r (Hand, 2015).	Probabilistic control via integrated model (Hunt, 1979) and bespoke coupling with external code (Bourgeois, 2005). Can implement user-defined models via open source code.	Prescribed schedules of lighting casual gains. Control based on required workplane illuminance.
IDA-ICE	Can implement user-defined models via standardised model input language (NMF).	Prescribed schedules.
IES-VE	Can implement user-defined models via proprietary model input language (APpro).	Prescribed schedules of light operation. Lighting levels can be daylight compensated.
Pleiades + Comfie (IZUBA énergies, 2016)	None.	Prescribed schedules of required workplane illuminance.
TRNSYS (TRNSYS17, 2012)	Can implement user-defined models via: 1) source code modification, 2) external function interaction, 3) proprietary model input language (W; Keilholz et al., 2009).	Prescribed schedules of lighting gains. Can co-simulate with external programs to provide daylighting control functionality.

Table 3: Summary of window operation modelling functionality.

Program	Stochastic functionality	Deterministic functionality
DeST	Opening/closing probability related to event or environment (Wang, 2014; Ren et al., 2014).	Prescribed schedules of window operation.
DOE-2.1E	Can implement user-defined models via external function interaction.	Control based on temperature.
EnergyPlus (U.S. Department of Energy, 2016)	Can implement user-defined models via: 1) (open) source code modification, 2) proprietary model input language (EMS), and 3) standardised co-simulation interfaces (FMI, BCVTB).	Control based on temperature, enthalpy, wind velocity and other metrics.
ESP-r (Hand, 2015).	Probabilistic control (Rijal et al., 2008). Can implement user-defined models via (open) source code modification.	Control based on temperature, wind velocity and other metrics.
IDA-ICE	Can implement user-defined models via standardised model input language (NMF; e.g. Anderson et al., 2013; D'Oca et al., 2014; Fabi et al., 2013a).	Prescribed schedules.
IES-VE	Can implement user-defined models via proprietary model input language (APpro).	Prescribed schedules of window operation. Window opening can be weather or comfort compensated.
Pleiades + Comfie (IZUBA énergies, 2016)	None.	Prescribed schedules of opening fraction.

TRNSYS (TRNSYS17, 2012)	Can implement user-defined models via: 1) source code modification, 2) external function interaction, 3) proprietary model input language (W; Keilholz et al., 2009).	Prescribed schedules of air change rate.

Table 4: Summary of HVAC operation modelling functionality.

Program	Stochastic functionality	Deterministic functionality
DeST	Operation probability related to event or environment (Wang, 2014; Ren et al., 2014; Wang et al., 2016).	Prescribed schedules of window operation.
DOE-2.1E	Can implement user-defined models via external function interaction.	Prescribed schedules of fan and thermostat settings.
EnergyPlus (U.S. Department of Energy, 2016)	Can implement user-defined models via: 1) (open) source code modification, 2) proprietary model input language (EMS), and 3) standardised co-simulation interfaces (FMI, BCVTB).	Prescribed schedules of HVAC availability/operation and thermostat settings.
ESP-r (Hand, 2015).	Can implement user-defined models via (open) source code modification.	Control by a variety of generally set point-based algorithms, based on weather, environment, etc.
IDA-ICE	Can implement user-defined models via standardised model input language (NMF; e.g. D'Oca et al., 2014; Fabi et al., 2013b).	Prescribed schedules.
IES-VE	Can implement user-defined models via proprietary model input language (APpro).	Prescribed schedules of HVAC component operation. Control by fuzzy logic and standard control algorithms.
Pleiades + Comfie (IZUBA énergies, 2016)	None.	Prescribed schedules of heating/cooling set points. Prescribed schedules of ventilation fraction.
TRNSYS (TRNSYS17, 2012)	Can implement user-defined models via: 1) source code modification, 2) external function interaction, 3) proprietary model input language (W; Keilholz et al., 2009).	Prescribed schedules of heating/cooling set points, capacities and humidification/dehumidification.

Table 5: Summary of other occupant action modelling functionality.

Program	Stochastic functionality	Deterministic functionality
DeST	None.	Prescribed schedules of appliance use.
DOE-2.1E	Probabilistic shading control when predefined criteria are met. Can implement user-defined models via external function interaction.	Prescribed schedules of casual gains and DHW use.

EnergyPlus (U.S. Department of Energy, 2016)	Can implement user-defined models via: 1) (open) source code modification, 2) proprietary model input language (EMS), and 3) standardised co-simulation interfaces (FMI, BCVTB).	Prescribed schedules of casual gains and DHW use. Control of shading by a variety of strategies, including schedules, trigger events and set points. Control of occupant clothing based on ASHRAE standard 55.
ESP-r (Hand, 2015).	Probabilistic control of fans (Rijal, 2008). Can implement user-defined models via (open) source code modification.	Prescribed schedules of other casual gains. Occupant-linked appliance use via bespoke coupling to external code (SHOCC; Bourgeois, 2005).
IDA-ICE	Can implement user-defined models via standardised model input language (NMF).	Prescribed schedules.
IES-VE	Can implement user-defined models via proprietary model input language (APpro)	Prescribed schedules of other casual gains. Gains can be occupancy compensated.
Pleiades + Comfie (IZUBA énergies, 2016)	None.	Prescribed schedules of other casual gains and DHW use.
TRNSYS (TRNSYS17, 2012)	Can implement user-defined models via: 1) source code modification, 2) external function interaction, 3) proprietary model input language (W; Keilholz et al., 2009).	Prescribed schedules of casual gains. Prescribed schedules of occupant clothing level, external work and metabolic rate.

Method

Data for 8 BPS programs are presented. These are: DeST v2.0, DOE-2.1E v124, EnergyPlus v8.3, ESP-r v12.3, IDA ICE v4.6, IES-VE 2016, Pleiades + Comfie v3.5.8.1 and TRNSYS 17 v5.3.0. Where the authors were not experienced in the use of these programs (IDA-ICE and IES-VE), information was sought from other parties with substantial knowledge of the program. This was preferred over investigation of these tools by the authors to take advantage of existing expertise in order to minimize the possibility of omitting or misunderstanding obscure or poorly documented functionality. Whilst every effort has been made to assess functionality and report information gathered as completely and accurately as possible, it is noted that there will always remain a possibility that some elements of functionality have been overlooked.

Information was gathered and recorded in the form of a questionnaire, strongly differentiating between deterministic (or prescribed) and stochastic models. Questions were divided into six modelling categories; occupant movement and/or presence, use of lights, use of windows, use of HVAC, other casual gains (e.g. small power), and any other occupant behaviors (e.g. shading). The same questions were asked for each of these areas:

- Does the BPS program include any stochastic model(s) of [modelling category]?
- If yes, please briefly describe the model(s).
- If yes, please give up to three references detailing each model.
- Please briefly describe any deterministic models of [modelling category] included in the BPS tool. Please also provide one reference detailing each model and/or its application.

Where information was gathered externally, the purpose of the activity was stated to each participant; to identify currently available occupant modelling functionality in a range of BPS programs. Also, the questionnaire was designed to minimize bias and maximize completeness in responses from external parties by making the questions specific and as closed-ended as possible.

The data was gathered to explore provision of models for each type of behavior, as well as a wider assessment of overall stochastic and deterministic occupant behavior modelling functionality. In a preamble, the questionnaire also requested that it be specified whether models were coded directly into the program, or could be implemented through a third party program or interface. This, along with the source code model for each program, allowed a further assessment of the ability to input user-defined occupant behavior models.

Table 6: Overview of stochastic functionality

	Stochastic models or potentially stochastic input capabilities for …				
Program	**Presence / movement**	**Lighting operation**	**Window operation**	**HVAC operation**	**Others**
DeST	Markov chain	Probabilistic control	Probabilistic control	Probabilistic control	None
DOE-2.1E	User-defined	User-defined	User-defined	User-defined	Probabilistic shading control, user-defined
EnergyPlus	User-defined	Scheduled probability, user-defined	User-defined	User-defined	User-defined
ESP-r	Probabilistic arrival and departure, user-defined	Proabilistic control, user-defined	Probabilistic control, user-defined	User-defined	Probabilistic fan control, user-defined
IDA-ICE	User-defined	User-defined	User-defined	User-defined	User-defined
IES-VE	User-defined	User-defined	User-defined	User-defined	User-defined
Pleiades + Comfie	None	None	None	None	None
TRNSYS	User-defined	User-defined	User-defined	User-defined	User-defined

Results

Tables 1-5 show summaries of functionality in each modelling category; Table 5 shows data for the “other casual gains” and “other occupant actions” categories combined. For convenience, a brief overview of available stochastic functionality is given in Table 6.

It is worth noting that where capability to input user-defined models is reported as stochastic functionality, this implies that the functionality can be used to input stochastic models, but is not limited to this; the funtionality is general and can also be used to input deterministic models.

Also, “external function interaction” refers to the capability to interact with user-defined functions written in a well established general coding language e.g. Fortran, C, C++; generally these will be compiled and imported as dynamic libraries. This is regarded as distinct from model input via an integrated coding interface or a dedicated language, which are more often interpretative.

Discussion

Firstly, the results of this investigation have confirmed the basic premise; whilst deterministic representation of occupant presence and behaviour is fairly consistent among BPS programs, stochastic functionality is far less homogenized.

Prescribed schedules and/or black-box control are the de facto standard in BPS tools; all of the programs surveyed have this deterministic functionality. There are minor variations between programs, for example some are limited to hourly resolution whilst others can handle sub-hourly resolution, and some have provision for control in aspects others do not, but the built-in deterministic functionality for representing occupants in BPS programs is in general quite similar. This commonality in prescriptive functionality is reflected in some standards, for example ASHRAE Standard 90.1 (ASHRAE 2013) includes suggestions for standard occupancy related profiles for various broad occupancy types (e.g. office). Such suggestions presuppose that schedule-based representation of occupants is available in whatever BPS program the user may be using.

On the other hand, stochastic representation of occupants is less ubiquitous. Whilst half of the programs reviewed include some built-in probabilistic modelling capability, the functionality is far from consistent. There are two broad types of built-in modelling capability: 1) hard-coded models with fixed operation probabilities and indicator variables, and 2) ability for users to input operation probabilities, or functions to calculate operation probabilities from indicator variables. The first type is only present in ESP-r of the programs reviewed, whereas the second type is present in DOE-2.1E and EnergyPlus in limited contexts, and in DeST to a wider extent. These two types of modelling capabilities exhibit a trade-off between prescriptive and flexible functionality; the former requires less input information but is less flexible, and the latter is more flexible but requires the user to specify opinions regarding operation probabilities.

Capacity to enter user-defined models is even more flexible but conversely requires more input from the user. This type of functionality is more common, but is enabled by a wide variety of input and coupling methods. Some programs such as IES-VE, EnergyPlus and TRNSYS have proprietary languages allowing users to input models, including stochastic models. This functionality is tailored to the specific program, and with a restricted remit it is likely that ample support is available for inexperienced users.

Other programs include in-built capacity to interact with external model exchange or co-simulation standards, for example IDA-ICE supports input of models according to that Neutral Model Format (NMF) standard, and EnergyPlus supports co-simulation according to the Functional Mockup Interface (FMI, 2016) standard. These standards seek to be general, but often must impose practical constraints such as I/O requirements in order to create a portable standard. Also the usefulness of such external standards is somewhat dependant on uptake and implementation, in this context among BPS programs. FMI is a relatively recent development, arising from the MODELISAR project begun in 2008, and NMF is an earlier attempt at a broadly similar goal that was first presented in 1989 (Sahlin & Sowell, 1989). NMF saw reasonable uptake in the 1990’s, for example the NMF Handbook (Sahlin, 1996) lists “translators” (i.e. interpreters to allow import of NMF models into building simulation tools) for 6 modelling programs including IDA and TRNSYS. However more recently NMF seems to have fallen out of favour for programs other than IDA-ICE; no research published after 2000 could be found that mentions use of NMF in any other context than IDA-ICE. FMI is currently seeing some uptake among BPS programs, being available in EnergyPlus, and seeing use in other programmable BPS programs for example TRNSYS (Elsheikh et al. 2013). An FMI implementation is also currently under development for ESP-r.

Finally, half of the programs reported (DOE-2.1E, EnergyPlus, ESP-r, TRNSYS) have capability for users to import external libraries, or modify the source code of the program itself. These methods generally require knowledge of classical programming languages such as Fortran and the various incarnations of C, and the processes involved in compiling such programs.

This may be viewed as a continuum from hard-coded models on the one hand, to completely arbitrary code input on the other hand. Moving from the former towards the latter generally requires greater effort, knowledge and technical skill from the user, but enables increased flexibility.

obFMU, as a Functional Mockup Unit (FMU), uses FMI to interface with BPS programs. As an emerging standard seeing increasing use, FMI is well placed to achieve significant penetration among BPS tools; this will be critical to the usefulness of obFMU. Lessons could perhaps be learned from precedent provided by NMF in this respect, though establishing the exact nature of this precedent is outwith the scope of this work.

In terms of the continuum of modelling functionality that has been established, obFMU seems to present a number of benefits. Provided that the FMI linkage is achieved through the interface of the BPS program, the functionality is likely to be reasonably intuitive and well supported, effectively removing much of the burden of coupling the model with the BPS program that is involved in many generalized model input methods. obFMU is distributed with a library of existing occupant behavior models, and being a program-independent standard model portability is high. Similarly then, in many cases this will also effectively remove much of the burden of implementing the model. However these benefits are dependent on existing implementations; clearly this is at odds with the flexibility of the method. In cases where researchers wish to implement novel models, or couple obFMU to programs which do not have in-built FMI functionality, or even couple in ways outwith the scope of the in-built FMI functionality, these benefits may not apply.

In this respect, rather than implicitly reducing the effort required to implement models, obFMU could act more as an aggregator of modelling effort. Assuming wide uptake of the FMI standard and an active dialogue between users and developers, the portability of obFMU could allow effort expended in one context to be leveraged in many others, effectively archiving functionality. For example, a new model could be implemented in obFMU, and this model could then in theory be shared with all other users of obFMU. Similarly, once a new input or output variable is implemented in obFMU, this would then be available for all other models to use.

However, there are caveats to the co-simulation approach. Firstly, FMUs and the FMI interfaces in BPS tools are defined by external standards, and BPS tool developers therefore lose some measure of control over their own program. obFMU and the interfaces in BPS tools must evolve in line with changes to the FMI standard; something that BPS tool developers are not likely to have much control over. This could result in some BPS programs becoming incompatible with obFMU by not implementing changes in their FMI interfaces on a time scale in line with changes to obFMU.

Also, co-simulation implicitly introduces increased computational overhead. It stands to reason that greater computational resources are required to run two programs as opposed to just one of them. This may limit the application of obFMU in complex models with short time steps for example, as the run-time or memory requirement of running obFMU at every time step, as well as the BPS simulation, may become prohibitive. Further investigation and precedent from use in practice could help to quantify this issue.

Finally, although the inputs and outputs to obFMU are clearly defined, the FMI implementation in each BPS program may vary. The models encapsulated in obFMU will be implicitly identical, but the details of how the outputs of these models are implemented in a simulation may differ between BPS programs. It remains the responsibility of the user to ensure that model outputs are applied in the appropriate way, for which documentation in terms of the models and the particular FMI implementation is likely to be key. This could be mitigated to some extent by ongoing cooperation between developers of different BPS tools.

Conclusions

This study has reviewed currently available occupant behavior modelling functionality in a variety of BPS tools, and established that whilst deterministic functionality is fairly well homogenised, stochastic functionality is not. Ability to enter user-defined models, which could include stochastic functionality, is fairly widespread, but the methods by which this is enabled are diverse which severerly compromises model portability. Given that current trends in the field are moving toward integrating stochastic modelling in BPS tools (Hong et al. 2016b), this highlights the widening gap between knowledge and implementation in the field of occupant behavior modelling.

obFMU provides a standard program-independent implementation of occupant behaviour models, which has potential to address this gap in available functionality; it has potential to contribute to fulfilling the documented need for "a standard framework to describe and model occupant behavior in buildings … to enable model exchange or reuse" (Yan et al. 2015).

However, there are a number of caveats and dependencies for it's usefulness. The following recommendations are made to mitigate the pitfalls, exploit the economies, and generally increase the value of obFMU:

- Every effort should be made to encourage further uptake of the FMI standard; the usefulness of obFMU is critical on implementation of FMI (and therefore capacity to co-simulate with obFMU) in as wide a variety of BPS tools as possible. In this respect further work could examine the lessons that could be learned from previous similar initiatives such as NMF.
- The benefits of obFMU may be realised to their full potential if the platform remains actively developed, and continues to implement and distribute the state-of-the-art in occupant behavior modelling.
- Cooperation and regular communication among developers of the FMI standard, and development teams of BPS tools, could help to mitigate compatibility issues that are incumbent in the co-simulation approach.
- Documentation of model inputs and outputs, particularly in terms of the implementations within obFMU, is likely to be critical to ensuring correct and effective use of obFMU.

Further work by the present authors will explore the performance and effects of obFMU in practice, both in terms of computational burden and the predictions from simulations.

Acknowledgements

The authors would like to thank all parties who provided information on BPS tools. Also the support of fellow participants of Annex 66 and the International Energy Agency's Energy in Buildings and Communities Programme is gratefully acknowledged.

References

Andersen, R. K., V. Fabi, J. Toftum, S. P. Corgnati and B. W. Olesen (2003). Window opening behavior modelled from measurements in Danish dwellings. *Building and Environment 69*, 101-113.

Baeten, R. and D. Saelens (2011). Integrating occupant behaviour in the simulation of coupled electric and thermal systems in buildings. *Proceedings of the 8th International Modelica Conference, Dresden, Germany*, 847-855.

BEST Directory (2016[?]). *Building Energy Software Tools*. Website last accessed 1/12/2016: http://www.buildingenergysoftwaretools.com/.

Bourgeois, D (2005). *Detailed occupancy prediction, occupancy-sensing control and advanced behavioural modelling within whole-building energy simulation*. Ph.D. thesis, Université Laval, Quebec, Canada.

Clarke, J. A., J. W. Hand, J. M. Kim, C. Ladha, P. Olivier, T. Roskilly, M. Royapoor, A. A. Samuel and D. Wu (2014). Pervasive sensing as a mechanism for improving energy performance within commercial buildings. *Proceedings of the 2nd Building Simulation and Optimization conference, London, England*.

Crawley, D. B., J. W. Hand, M. Kummert and B. T. Griffith (2008). Contrasting the capabilities of building energy performance simulation programs. *Building and environment 43*(4), 661-673.

D'Oca, S., V. Fabi, R. K. Andersen and S. P. Corgnati (2014). Effect of thermostat and window opening occupant behavior models on energy use in homes. *Building Simulation 7*, 683–694.

Elsheikh, A., E. Widl, P. Pensky, F. Dubisch, M. Brychta, D. Basciotti and W. Müller (2013). Modelica-enabled rapid prototyping via TRNSYS. *Proceedings of the 13th International Conference of the International Building Performance Simulation Association, Chembéry, France.*

EQUA Simulation AB (2016). *Highlights in [version] 4.6*. Website available online: http://www.equa.se/en/ida-ice/what-is-new/previous-versions#v46.

Fabi, V., S. D'Oca, T. Buso and S. P. Corgnati (2013a). The influence of occupant's behavior in a high performing building. *Proceedings of the 7th Mediterranean Congress of Climatization, Istanbul, Turkey.*

Fabi, V., R. V. Andersen and S. P. Corgnati (2013b). Influence of Occupant's Heating set-point preferences on Indoor Environmental Quality and Heating Demand in Residential Buildings. *HVAC&R Research Journal 19*(5), 635-645.

Feng, X., D. Yan and T. Hong (2015). Simulation of occupancy in buildings. *Energy and Buildings 87*, 348-359.

FMI (2016[?]). *FMI: Functional Mock-up Interface*. Website last accessed 1/12/16: https://www.fmi-standard.org/.

Gilani, S., W. O'Brien, H. B. Gunay and J. S. Carrizo (2016). Use of dynamic occupant behavior models in the building design and code compliance processes. *Energy and Buildings 117*, 260-271.

Hand, J. W (2015). *Strategies for deploying virtual representations of the built environment, aka the ESP-r cookbook*. ESRU, University of Strathclyde, Glasgow.

Hoes, P., J. L. M. Hensen, M. G. L. C. Loomans, B. de Vries and D. Bourgeois (2009). User behavior in whole building simulation. *Energy and Buildings 41*, 295-302.

Hong, T., S. D'Oca, W. Turner and S. C. Taylor-Lange (2015a). An ontology to represent energy-related occupant behavior in buildings. Part I: Introduction to the DNAs Framework. *Building and Environment 92*, 764-777.

Hong, T., S. D'Oca, S. C. Taylor-Lange, W. Turner, Y. Chen and S. Corgnati (2015b). An ontology to represent energy-related occupant behavior in buildings. Part II: Implementation of the DNAS framework using an XML schema. *Building and Environment 94(1)*, 196-205.

Hong, T., Y. Chen, Z. Belafi and S. D'Oca (2016a). Occupant behavior models: Implementation and representation in building performance simulation programs. *Building Performance Simulation*, submitted.

Hong, T., S. C. Taylor-Lange, S. D'Oca, D. Yan and S. P. Corgnati (2016b). Advances in research and applications of energy-related occupant behavior in buildings. *Energy and Buildings 116*, 694-702.

Hong, T., H. Sun, Y. Chen, S. C. Taylor-Lange and D. Yan (2016c). An occupant behavior modeling tool for co-simulation. *Energy and Buildings 117*, 272-281.

Hunt, D. R. G (1979). The use of artificial lighting in relation to daylight levels and occupancy. *Building and Environment 14*(1), 21-33.

IZUBA énergies (2016[?]). Bibliothèque / Scénarios / Scénarios hebdomadaires. Website available online: http://docs.izuba.fr/fr/index.php/Biblioth%C3%A8que_/_Sc%C3%A9narios_/_Sc%C3%A9narios_hebdomadaires [in french].

Jang, H. and J. Kang (2016). A stochastic model of integrating occupant behaviour into energy simulation with respect to actual energy consumption in high-rise apartment buildings. *Energy and Buildings 121*, 205-216.

Keilholz, W., P. Sette and E. H. Soussi (2009). *The W programming language User & Reference Manual*. Centre Scientifique et Technique du Bâtiment, Paris, France.

Ren, X., D. Yan and C. Wang (2014). Air-conditioning usage conditional probability model for residential buildings. *Building and Environment 81*, 172-182.

Rijal, H. B., P. Tuohy, M. A. Humphreys, J. F. Nicol, A. Samuel, I. A. Raja and J. Clarke (2008). Development of adaptive algorithms for the operation of windows, fans and doors to predict thermal comfort and energy use in Pakistani buildings. *ASHRAE Transactions 114*(2), 555–573.

Sahlin, P. and Sowell, E. F. (1989) A Neutral Format for Building Simulation Models. *Proceedings of the 2nd IBPSA Building Simlulation conference 1989, Vancouver, Canada.*

Sahlin, P. (1996). *NMF Handbook: An Introduction to the Neutral Model Format.* Technical Report.

TRNSYS17 (2012). *Volume 5 Multizone Building modeling with Type56 and TRNBuild*. Technical report.

U.S. Department of Energy (2016). *EnergyPlusTM Version 8.5 Documentation: Input Output Reference*. Document available online: https://energyplus.net/sites/all/modules/custom/nrel_custom/pdfs/pdfs_v8.5.0/InputOutputReference.pdf.

Virote, J. and R. Neves-Silva (2012). Stochastic models for building energy prediction based on occupant behavior assessment. *Energy and Buildings 53*, 183-193.

Wang, C., D. Yan and Y. Jiang (2011). A novel approach for building occupancy simulation. *Building Simulation 4*(2), 149-167.

Wang, C. (2014). *Simulation research on occupant energy-related behaviors in building*. Ph.D. thesis, Tsinghua University, China.

Wang, C., D. Yan and X. Ren (2015). Modeling individual's light switching behavior to understand lighting energy use of office building. *Proceedings of the Applied Energy Symposium and Summit 2015, Fuzhou, China.*

Wang, C., D. Yan, H. Sun and Y. Jiang (2016). A generalized probabilistic formula relating occupant behavior to environmental conditions. *Building and Environment 95*, 53-62.

Yan, D., W. O'Brien, T. Hong, X. Feng, H. B. Gunay, F. Tahmasebi and A. Mahdavi (2015). Occupant behavior modelling for building performance simulation: Current state and future challenges. *Energy and Buildings 107*, 264-278.

Zhou, X., T. Hong and D. Yan (2014). Comparison of HVAC Modeling in EnergyPlus, DeST, and DOE-2.1E. *Building Simulation 7*, 21-33.

Zhu, D., T. Hong and D. Yan (2013). A Detailed Comparison of Three Building Energy Modeling Programs: EnergyPlus, DeST, and DOE-2.1E. *Building Simulation 6*(3), 323-335.