Contents lists available at ScienceDirect

Energy and Buildings

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

Simulation of occupancy in buildings

Xiaohang Feng[a], Da Yan[a], Tianzhen Hong[b,*]

[a] School of Architecture, Tsinghua University, Beijing 100084, China
[b] Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94720, USA

ARTICLE INFO

Article history:
Received 19 July 2014
Received in revised form 22 October 2014
Accepted 25 November 2014
Available online 4 December 2014

Keywords:
Building simulation
Co-simulation
Occupancy
Occupant behavior
Software module
Stochastic modeling

ABSTRACT

Occupants are involved in a variety of activities in buildings, which drive them to move among rooms, enter or leave a building. In this study, occupancy is defined at four levels and varies with time: (1) the number of occupants in a building, (2) occupancy status of a space, (3) the number of occupants in a space, and (4) the space location of an occupant. Occupancy has a great influence on internal loads and ventilation requirement, thus building energy consumption. Based on a comprehensive review and comparison of literature on occupancy modeling, three representative occupancy models, corresponding to the levels 2–4, are selected and implemented in a software module. Main contributions of our study include: (1) new methods to classify occupancy models, (2) the review and selection of various levels of occupancy models, and (3) new methods to integrate these model into a tool that can be used in different ways for different applications and by different audiences. The software can simulate more detailed occupancy in buildings to improve the simulation of energy use, and better evaluate building technologies in buildings. The occupancy of an office building is simulated as an example to demonstrate the use of the software module.

1. Introduction

Occupants are involved in a variety of activities in a building, such as working in their offices, communicating with other occupants, meeting in a conference room, or going to a restroom. Activities drive occupants' movement among rooms and in and out of a building. Occupancy here is defined as occupied status or number of occupants at four levels varied with time: (1) the number of occupants in a building, (2) a space is occupied or not, (3) the number of occupants in a space, and (4) in which space an occupant is located.

Occupancy has great impact on building energy consumption and is the basis of building simulation with tools such as EnergyPlus [1] and DeST [2]. It is also a key to occupant behavior modeling, e.g. window opening and closing or HVAC systems turning on and off. With the global trend toward low energy buildings, occupancy based controls are being used to reduce energy use in buildings, such as air-conditioning [3], ventilation [4], and lighting [5].

Some devices are controlled by occupancy sensors and perform independently of occupant behavior decision making. For example, a lighting system installed with an occupancy sensor in a room can be turned off if the sensor signals that the room is unoccupied, and it can be turned on if the sensor sends an occupied signal. A typical 30% energy savings can be achieved with occupancy sensors to control lighting [6]. Lo and Novoselac [7] simulated the energy consumption with occupancy control in an open office, and found that a 30% cooling energy reduction is achieved when the thermostat setting of the unoccupied zones were reset to have a higher temperature than that of the occupied zones.

Occupant behavior varies by individuals. Yun et al. [8] found that the lighting use patterns are significantly related to the occupancy patterns in offices. An investigation of residences in Kuwaiti conducted by Al-Mumin et al. [9] suggests that the air-conditioning (AC) thermostat settings of surveyed residences vary from below 19 °C to above 25 °C. Brager et al. [10] states that thermal sensations were broadly distributed in the same way from both the warm and cool season surveys due to individual preferences. This may lead to different window operations. Some occupants may prefer the window to be closed, while others to be open, even under the same room thermal conditions. From this perspective, it is important to know who is in a space at a particular time in order to determine operations of energy-related devices for comfort.

Some systems are demand controlled, i.e. the number of occupants in a space determines the operation of such systems.

* Corresponding author. Tel.: +1 510 4867082; fax: +1 510 4864089.
E-mail addresses: fengxh12@mails.tsinghua.edu.cn (X. Feng), yanda@tsinghua.edu.cn (D. Yan), thong@LBL.gov (T. Hong).

http://dx.doi.org/10.1016/j.enbuild.2014.11.067
0378-7788/© 2014 Elsevier B.V. All rights reserved.

Ventilation is always required for an occupied space to maintain proper indoor air quality. As more occupants fill a space, the more ventilation air is needed in that space. It has been demonstrated that energy savings by using occupancy controlled ventilation can be achieved, by reducing the average ventilation rate while keeping an acceptable indoor climate [11]. In this way, better control of indoor pollutant concentrations while at the same time lower energy use and peak energy demand, can be achieved [12].

Some devices, like laptop computers, are often attached to individual occupants. Portable devices move together with the occupants from space to space and consume energy or generate heat gains. Since most occupant behavior patterns are influenced by occupancy. Therefore, simulating occupancy becomes fundamental for occupant behavior research. Current building energy simulation usually treats occupancy as deterministic daily profiles, varying between 0 and 1, presenting no occupancy and full occupancy. Fig. 1 shows typical occupancy schedules (for the whole building) on weekdays, Saturday, and Sunday and holidays for office buildings, which are part of the 16 prototype buildings developed by the United States Department of Energy, covering 80% of the commercial building floor area in the United States [13].

In such profiles, although the diversity between workdays and holidays is considered, all workdays, Saturdays and Sunday and holidays in a building are described with identical schedules, respectively. As shown in Fig. 1, the number of occupants in the building reaches 10% of full occupancy at 7 am, and increases gradually to a maximum value of 95% at 9 am. It drops to 50% during lunch time and returns to 95% after lunch. Occupants start to leave the building at 5 pm and finally the building restores to no occupancy status at 12 am. When these homogeneous occupant profiles are used, each space will have same or very similar load profiles, thus no diversity is considered. Duarte et al. [14] collected long-term data to show variations of occupancy diversity factors in private offices for time of day, day of the week, holidays, and month of the year, which show differences as much as 46% from those currently recommended by ASHRAE Standard 90.1 2004 energy cost budget guideline, a document referenced by energy modelers regarding occupancy diversity factors for simulations. The simplification of homogeneous schedules also does not capture actual system performance. For example a variable air volume (VAV) system would perform very well without reheat penalty because spaces have similar load profiles, while in reality the diversity of loads usually cause reheat because some spaces call for high cooling while others call for very low cooling [15].

In fact, occupancy in a building is stochastic both in time and in space. Thus a discrepancy occurs between the actual and simplified occupancy profiles. When different occupancy profiles are used to simulate building energy consumption, the deterministic profiles may be incapable of estimating the peak energy consumption when special events occur. Moreover, profiles may overestimate the peak load of spaces in the same system partition as they reach the maximum occupancy simultaneously, which is often not realistic due to stochastic nature of occupant behavior in buildings. To describe occupancy more realistically, various models on occupancy simulation have been developed in literature. Energy modeling programs can use more accurate and dynamic occupancy schedules by integrating these occupancy models.

In this study, four levels of occupancy models for various applications were identified. The reviewed and selected existing occupancy models were used in the development of software architecture to integrate these models. Finally the application of the software tool, to generate occupancy schedules which better represent the spatial and temporal diversity of occupancy in buildings, was demonstrated.

2. Review of occupancy models

Four types of occupancy models are categorized according to the problems they try to address:

(1) Building level and number of occupants, which addresses how many occupants are in a building at a particular time.
(2) Space level and occupied status, which addresses whether or not a space is occupied at a particular time. This is the information required by devices like lighting, with less emphasis on the number of occupants and more focus on the status of whether a space is occupied to determine the action of lighting devices.
(3) Space level and number of occupants, which addresses how many occupants are in a space at a particular time. One example is the demand controlled system, which requires information as detailed as the number of occupants, regardless of which occupants are in the space to determine the demand and operation.
(4) Occupant level, which addresses individual tracking and is the most detailed level.

The problem to address is in which space an occupant is at a particular time. As mentioned before, occupant behavior is different among individuals, thus the operation or performance of systems in a space is significantly based on the individual who occupies the space if one has access to the control of these systems. Each of the four level models addresses a specific occupancy application, and the information required is different by devices or operations.

Literature on occupancy modeling is reviewed and models are categorized into the four levels. At the first level, building level, unfortunately none is found in the building energy domain. Other domains, like emergency evacuation, may be more interested at this level.

2.1. Occupancy model, Level 2: occupancy status of a space

Several relevant papers are found for the second level, as listed in Table 1.

Yu [16] applies genetic programming (GP) algorithm to learn the behavior of an occupant in a single-person office based on motion sensor data. The developed rules provide prediction accuracies of 80–83% on testing data from five different offices. Six variables are provided for GP to learn the occupancy rules, i.e. day, hour, minute, the length of time the occupant spent in the state prior to the previous state, and the length of time the occupant has been in the office since the first arrival of the day. Wang et al. [17] examines the statistical properties of occupancy in single-person offices of a large office building. A probabilistic model to predict and simulate occupancy in single-person offices is proposed. The states of an occupant are divided into occupancy and vacancy. The interval lengths are modeled as exponentially distributed random variables. Chang and Hong [18] statistically analyses information collected from 200 cubicle offices on three floors of a commercial office building. A mathematical model, to describe the occupancy patterns, including the probability distributions of the number of absences and absence duration, is developed with three key elements: the cumulative distribution function (CDF) of the number of daily absences, CDF of each absence duration, and the probability distribution function (PDF) of the start time of each absence. These models focus on the occupancy status of a single-person office or cubical, and are not able to extend to the case where multiple occupants are present in an office since the probability distribution is totally changed, so the models have a common limitation that they are only applicable to one occupant in a space.

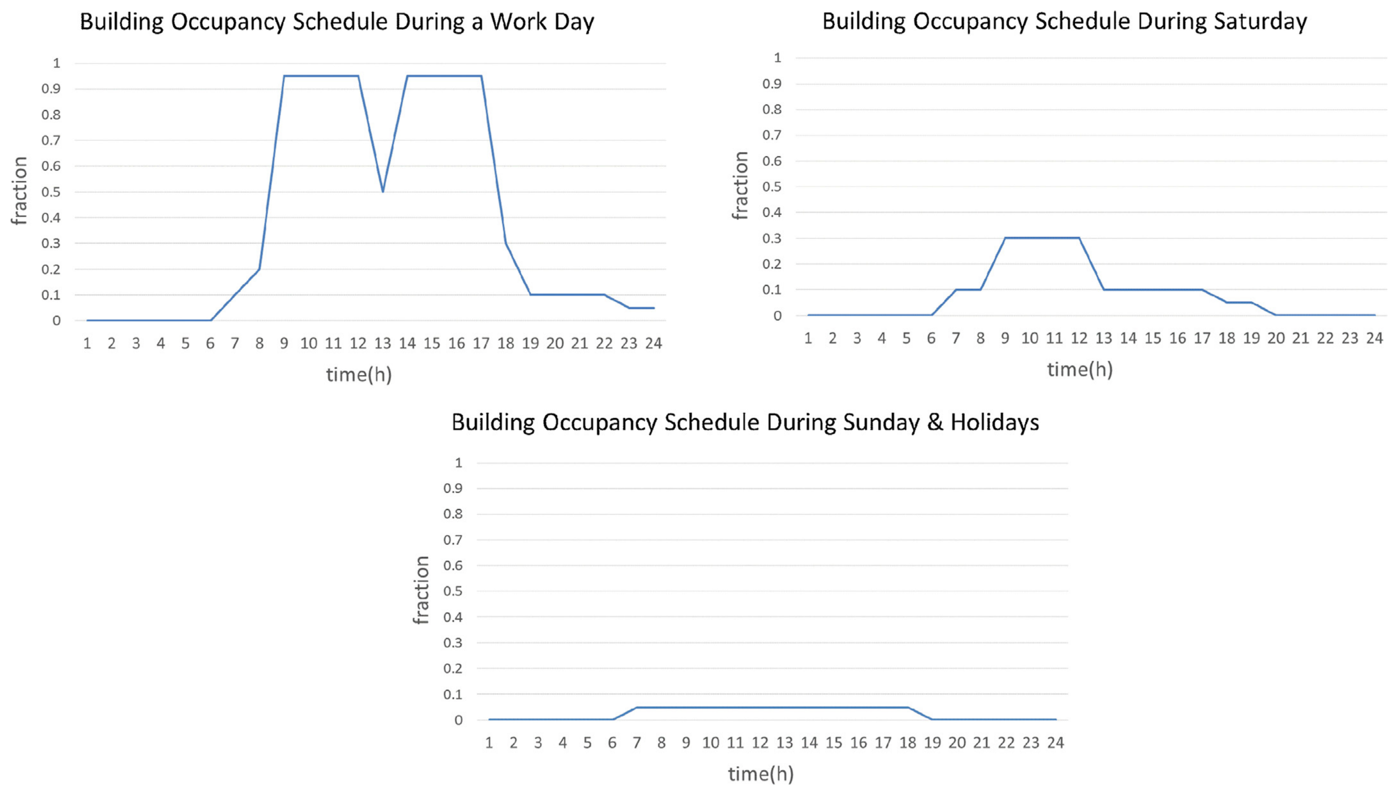

Fig. 1. Building occupancy schedules on a typical workday, Saturday and Sunday and holidays.

Table 1
Literature on occupancy status of a space.

Authors	Keywords	Pros	Cons

Yu [16]	Occupancy model;
	Genetic programming;
	Time variables
		Relatively accurate;
		Existing algorithm
			Field data required;
			Less accurate for departure

Wang et al. [17]	Commercial buildings;
	Occupancy properties;
	Single person offices
		Matches with observed;
		time varying;
			Intervals not fit well

Chang et al. [18]	Occupancy pattern; Statistical analysis	Classification of occupants; Easy to implement	Patterns not validated

Table 2
Literature on number of occupants of a space.

Authors	Keywords	Pros	Cons

Goldstein et al. [19]	Space layout;
	Occupant model;
	Cost function
		Interdependent;
		Classification
			Layout needed;
			Numerous schedules

Page et al. [20]	Presence model;
	Stochastic process;
	Markov chain
		No limitations to scales;
		Easy to implement
			Long-term monitoring

Goldstein et al. [21]	Occupant interaction;
	Occupant schedule;
	Customization
		Interdependent;
		Classification
			Dependent on real
			schedule extensively

2.2. *Occupancy model, Level 3: number of occupants in a space*

There are also several relevant papers on the third level, as listed in Table 2.

Goldstein et al. [19] shows space layout influences the selection of individuals who participate in an activity, and the location where the activity occurs. Participants and locations are randomly selected based on probabilities derived from cost functions. When a new activity is generated, participants are selected from all occupants based on probabilities from cost value, after which the location of the activity is selected. Page et al. [20] describes an algorithm for the simulation of occupant presence, to be later used as an input for occupant behavior models within building simulation tools. By considering occupant presence as an inhomogeneous Markov chain interrupted by occasional periods of long absence, the model generates a time series of the state of presence (absent or present) of each occupant of a space, for each space of a building. The IFM (inverse function method) is applied to generate a sample of events from a given probability distribution function. Goldstein et al. [21] introduces personas, fictional individuals used in the field of human–computer interaction, into the simulation of building performance. It extends the previous method [22] to allow

Table 3
Literature on locations of occupants in buildings.

Authors	Keywords	Pros	Cons
Wang et al. [24]	Occupant movement; Stochastic process; Markov chain	No constraint with scales; Parameters easily decided	Large matrices to store; Arithmetic speed problem
Nassar and Elnahas [25]	Occupant movement; Random walks; Architectural design	Randomness of walking; No constraint with scales;	Detailed layout needed; Activities not reflected
Liao and Barooah [26]	Agent-based model; Commercial buildings	Related with former state; Improving Page's model	Much information to store for an instance
Tabak [27]	USSU system; Office building	Cover many activities; More realistic in events	Distance between every space needed

occupants to interact with one another. Both occupant interaction and behavior customization were achieved via the assignment of weighting coefficients to activities used for model calibration. Models at this level commonly study occupancy in a room by estimating each occupant, who is attached with some properties or may have interaction with other occupants. Richardson et al. [23] presents a method for generating realistic occupancy data for UK households, based upon surveyed time-use data describing what and when people do. The approach he presented generates statistical occupancy time-series data and the number of occupants that are active within a house at a given time, which is important to model the sharing of energy use. The model has already been implemented and is freely available.

2.3. Occupancy model, Level 4: space location of an occupant

More models are found to address the last occupant level problem, i.e. individual tracking, as listed in Table 3.

Wang et al. [24] proposes an approach for building occupancy simulation based on the Markov chain. By using the Markov chain method to simulate this stochastic movement process, the model can generate the location for each occupant and further aggregated as the zone-level occupancy for the whole building. Each occupant is attached with a homogenous Markov matrix to simulate the stochastic movement process. Nassar and Elnahas [25] presents a basic measure for analyzing the design of building spaces in terms of space accessibility. The proposed measure, namely the random access measure (RAM), is presented as a useful and simple design analysis technique that relates to occupant movement as well as to space topology. Provided the starting point of an occupant, the model uses the random accessibility measure to simulate one's position after a time step. Liao and Barooah [26] develops an agent-based model to simulate the behavior of all occupants in a building, and extract reduced-order graphical models from Monte-Carlo simulations of the agent-based model. The model, named mixed agent-based rules model (MARM), consists of a number of modules for each agent i. Knowing Pi,j (k), the probability of agent i occupying node j at time k, an acceleration rule and a damping rule are used to mimic the behavior. Finally access profile is associated with each agent. Tabak [27] aims to develop a system that can be applied for analyzing and evaluating the space utilization of a building for any given organization. First, the activity location choice algorithm creates a choice set consisting of all relevant choices. Next, the algorithm calculates, for each activity location in the choice set, the distance to the location of the previous activity in one's activity. Finally, the activity location associated with the shortest distance is chosen and allocated to the activity. Models at this level are the most detailed, providing the information of spaces occupied by each occupant.

Since occupant activity is complicated, simplification is required during modeling. Some models are still too complicated to implement or need a large amount of inputs which tend to be simplified. Several comments on occupancy modeling are summarized from the literature review: (1) occupancy models often treat an occupant as an object, to study the presence or movement. This is very natural and makes sense. (2) No relevant literature on the building level occupancy modeling is found in the domain of building energy simulation. Perhaps other domains like emergency evacuation may be extended to review models at this level in the future. (3) There are no direct models on space level occupancy modeling except single-occupancy spaces, instead the number of occupants or occupancy status is found by aggregating the state or location of individual occupants. However, information on the number of occupants can be obtained without calculation of each occupant if direct models are available, which may save calculation time and have a certain degree of accuracy. (4) There are several detailed tracking models at the occupant level, but they usually require lots of user inputs which are sometimes hard to get. The challenge lies in how to convert these inputs to a small set of parameters which can be understood by a respondent.

3. Occupancy models selected for implementation

Occupancy models are supposed to support the design and evaluation of building performance by generating more realistic occupant schedules. According to this criterion, a model is considered good if it has a limited number of input parameters, which can easily be determined by surveys or a small amount of measured data. Often, these models do not require detailed building layout or specific information. Based on a comprehensive comparison of the advantages and disadvantages of these models, three of the occupancy models, each representing one of the three levels (Levels 2–4) are selected to implement in a software module.

3.1. Occupancy model, Level 2: occupancy state of a space

Chang's model [18] is selected to simulate the occupancy state of a space, i.e. being present or absent at a certain time in an office cubical. It is developed with three key elements: (1) the cumulative distribution function (CDF) of the number of daily absences, (2) the CDF of each absence duration, and (3) the probability distribution function (PDF) of the start time of each absence.

Once the profile of occupancy patterns are determined, the occupancy schedules can be generated by the following steps: (1) generate a uniform distribution of random numbers between 0 and 1 for the CDF of the number of daily absences to decide the number of daily absences, (2) generate a uniform distribution of random numbers between 0 and 1 for the CDF of the absence duration for each absence, (3) generate a uniform distribution of random numbers to calculate the start time of each absence. Note that the distribution of absence start time varies among occupancy patterns and is not always uniform.

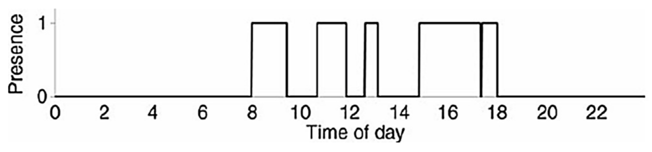

Fig. 2. Simulated schedule of occupant presence during a day.

Fig. 2 shows the sample simulation result of an occupant, where the number 1 means that the occupant is present at that time, while number 0 indicates absence.

3.2. *Occupancy model, Level 3: number of occupants in a space*

Page's model [20] is selected to simulate number of occupants in a space. It assumes that the presence of each occupant is independent and the probability that an occupant is present only depends on whether one was present at the previous time step.

The model takes a profile of the probability of the presence over a typical week and the parameter of mobility to determine the probability distribution of presence. For example, T01, indicates the probability of being present now when being absent at the previous time step, and T10, indicates the probability of being absent now when being present at the previous time step. The inverse function method (IFM) is used to determine the next state of presence. In a special case where long absence occurs, given the probability of starting a period of long absence is determined by the IFM. If there is a long absence, then the length of the absence is determined by the distribution of the duration of periods of long absences with the same method, during which period the occupant is considered to be absent.

3.3. *Occupancy model, Level 4: space location of an occupant*

Wang's model [24] is selected to generate the space location of each occupant and the space-level occupancy for the whole building. The core concept is the use of Markov chain. A Markov chain is a matrix with elements representing probabilities. The row index denotes the previous occupancy states, while the column index denotes the current occupancy states. The elements denote the probability of transferring from state [row index] to state [column index]. Here is a simple example:

$$\begin{bmatrix} 0.2 & 0.3 & 0.5 \\ 0.3 & 0.4 & 0.3 \\ 0.1 & 0.6 & 0.3 \end{bmatrix}$$

The element 0.5 in this matrix represents that if an occupant is in state 1 (Row 1), there is a probability of 0.5 for (s)he to transfer to state 3 (Column 3). This model is based on events, which use matrices to represent properties attached to an event and an occupant. Typical events defined in an office building include going to work, going for lunch, returning back from lunch, leaving work, and meeting. Some parameters are defined to describe the events in a simple way. For example, the range of arrival times and the average arrival time are used to describe the event "going to work".

Once the parameters of an event are known, a random number between 0 and 1 is generated to decide what event is taking place and the Markov chain is updated correspondingly. For each occupant, the new location of an occupant is determined by comparing a generated random number with the probabilities in the Markov chain.

4. Software architecture

Based on the comprehensive review of the existing occupancy models in literature, a software module was developed to include three representative occupancy models which can be used to simulate various occupancy levels. The software architecture of the occupancy module was developed to be object oriented, flexible and extensible. To the best of our knowledge no existing software tools are available which satisfy the cohesive integration of the three representative occupancy models. Fig. 3 shows the high-level software architecture with the major software classes.

The classes are designed to model real-world objects, encapsulate implementation details, and streamline parameter exchange [28]. Classes for a building, a room, an occupant, an event and a schedule are explicitly defined, with some properties to describe the object and methods to exchange parameters or implement the algorithms. CBuildings contains a collection of instances of CBuilding, where there is an instance of CSchedule representing the number schedule of number of occupants in the building.

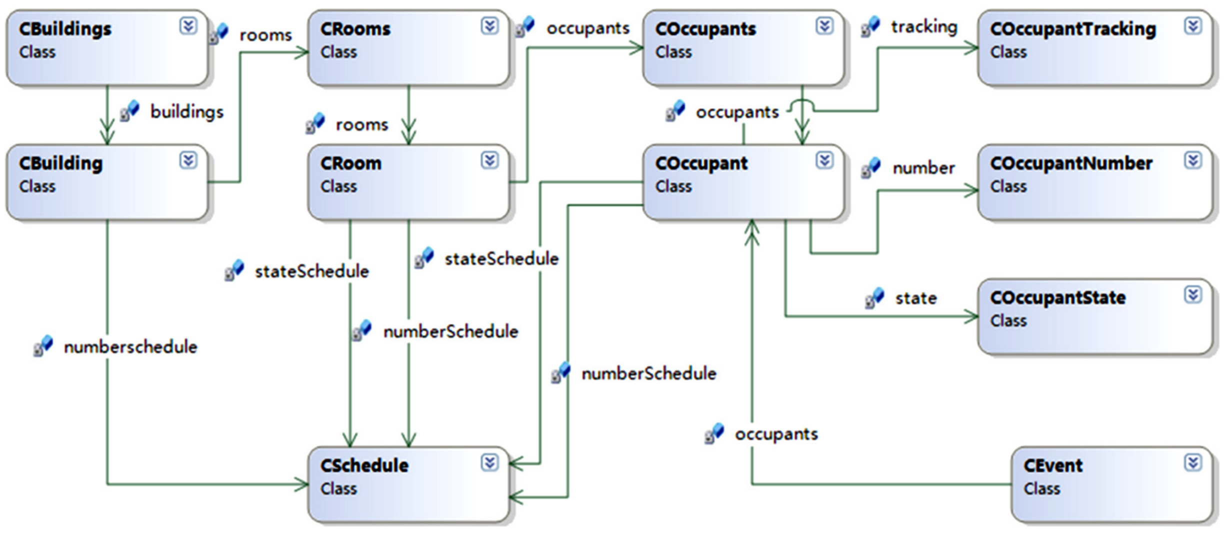

Fig. 3. Software architecture of the occupancy module.

Table 4
Properties and methods of the CBuilding class.

	Members	Comments
Properties	ID	An ID is required to uniquely refer to a building
	Type	Building type e.g. office or residential building
	Number	Number of rooms in the building
	Schedule	Time schedule of number of occupants in the building
	Rooms	Rooms in the building
Method	Calculate occupancy	If room-level is known, building-level can be deduced

Table 5
Properties and methods of the CRoom class.

	Members	Comments
Properties	ID	An ID is required to uniquely refer to a room
	Building	The building the room belongs to
	Type	Room type e.g. meeting room, office, corridor, etc.
	State schedule	Time series of occupancy state (occupied or not) of the room
	Number schedule	Time series of occupancy number of the room
	Occupants	Occupants in the room
Method	Calculate occupancy	If occupant-level is known, room-level can be deduced

Table 6
Properties and methods of the COccupant class.

	Members	Comments
Properties	ID	An ID is required to uniquely refer to an occupant
	Room	The room the occupant belongs to
	Type	The job category of an occupant
	State schedule	Time series of occupancy state (present or not) of the occupant
	Number schedule	Time series of locations (in which room) of the occupant
Method	Set events	To specify the activity conducted by the occupant
	Simulate state	To calculate time series of presence of the occupant
	Simulate location	To calculate time series of the room occupied by the occupant

Table 7
Properties and methods of the CEvent class.

	Members	Comments
Properties	ID	An ID is required to uniquely refer to an event
	Type	Type of the event, e.g. meeting, going to work, leaving from work
	Objected Room	The room in which the event takes place
	Time Range	When the event starts and ends
	Occupants	Occupants who are involved in the event

Table 8
Properties and methods of the CSchedule class.

	Members	Comments
Properties	ID	An ID is required to uniquely refer to a schedule
	Type	Type of the schedule, i.e. what the schedule represent
	Value	Time series of rooms or states
Method	Get attributes	To get statistical information, e.g. the maximum number of occupants

Similarly, CBuilding has an instance of CRooms, a collection of instances of CRoom, which contains two instances of CSchedule representing both occupancy state and number in a room. COccupants is a collection of instances of COccupant, which contains various modules represented as classes to implement different occupancy models. To calculate occupancy for each building, room or occupant, we can simulate each one in the collection successively. The class CEvent contains information to simulate what an event is taking place at a particular time and what occupants are involved. Such information are inputs to COccupants. The other three classes, COccupantState, COccupantNumber and COccupantTracking refer to the three selected occupancy models mentioned before. An additional new occupancy model is easy to integrate in this module by creating a new class to implement such model without the need to change existing classes. The major software classes are described as follows:

(1) The CBuilding contains information required by the module to describe the main properties of a building and methods to calculate occupancy at the building level. Table 4 presents the properties and methods of the CBuilding class.
(2) The CRoom contains information required by the module to describe the main properties of a room and methods to calculate occupancy at the room level. Table 5 presents the properties and methods of the CRoom class.
(3) The COccupant contains information required by the module to describe the main properties of an occupant and methods

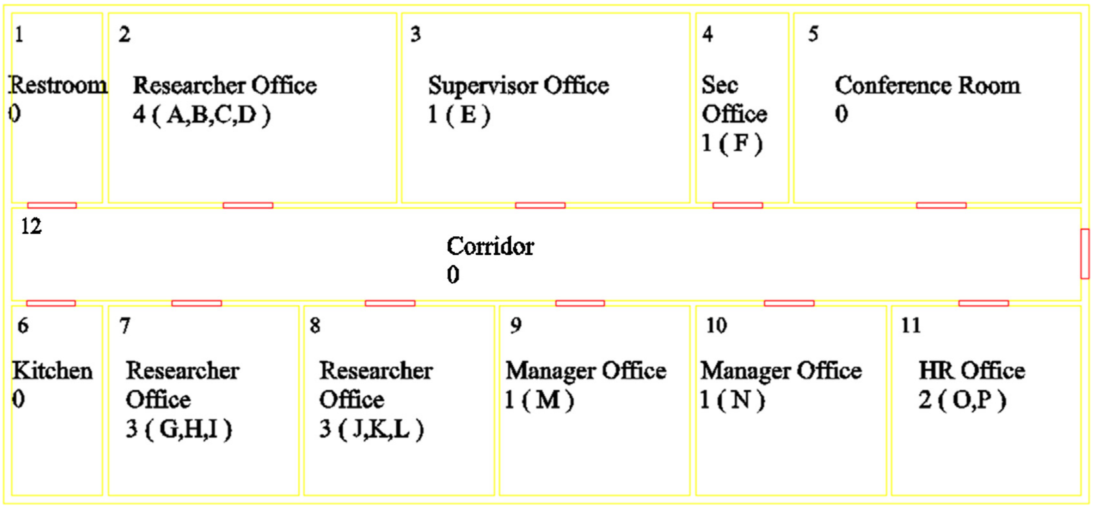

Fig. 4. Floor plan of a single-story office building.

to calculate occupants' state or location. Table 6 presents the properties and methods of the COccupant class.

(4) The CEvent contains the description of main properties of an event, includes start time, duration, occupants involved, etc. Table 7 presents the properties and methods of the CEvent class.

(5) The CSchedule contains the description of a time series to represent the number of occupants by time or occupancy state of a space by time. Table 8 presents the properties and methods of the CSchedule class.

5. Applications

The software module can be used in three ways. First, it can be run stand-alone, as an executable file to pre-calculate occupancy schedules. Secondly, it can be integrated into building energy modeling programs as a dynamic link library (DLL). Thirdly, it can be used via co-simulation with other programs. To call functions in the DLL, an energy modeling tool has to rely on code to trigger the calls and handle the inputs and outputs from the calls.

The major part of the module is implemented in a DLL, which can be called by other simulation programs. A stand-alone executable file is also developed to generate the occupancy schedule as input to simulation tools, which is a pre-simulation process. To integrate with current simulation tools, the concept co-simulation is introduced. Sections 5.1 and 5.2 go further into the details and examples of using the occupancy module during pre-simulation or co-simulation.

5.1. Pre-simulation

An example was built to demonstrate the usage and results of the developed occupancy module. Fig. 4 shows a floor plan of a single-story office building. Different types of rooms and occupants were defined including (1) eight offices, usually occupied by staff and (2) a restroom, a corridor, a kitchen and a conference room, which were usually unoccupied or occupied by shorter durations of time. In Fig. 4, the numbers below the room label denotes the number of occupants in the room. For example, four researchers are working, labeled by B, G, K and N. The floor plan shows a snapshot of the occupant information, while the actual occupants are moving around and the number of occupants in each room varies with time.

To simulate the occupancy in this building, some inputs are required. Taking the occupant tracking model for example, the number of occupants in each room must be specified. Information about meetings are required for the conference room. In this example, two kinds of meetings are set, random and scheduled meetings. Meetings may be randomly held between 9:00 am and 11:30 am. A scheduled meeting is held between 2:00 pm and 3:00 pm every day. Other information besides the meeting time range is also set, as shown in Table 9.

Different types of occupants also have different properties of moving because of individual preferences or vocations. For example, a supervisor tends to go outside often to attend meetings or other activities, thus the proportion of time he stays in his own office may be shorter than other employees. On the contrary, a researcher, who tends to spend a lot of time in the office doing research work, will have a larger office occupied period. In this example, typical inputs for each kind of occupants are assumed, as shown in Table 10. *P* represents the percentage of time an occupant is in one of the four space types. Surveys may be conducted and field data can be used as inputs to simulate occupancy in the future.

Once these inputs are set, the software module can calculate the locations of each occupant at a particular time, and the total number of occupants in a room or building varying with time can be aggregated from the location of individual occupants.

As the calculation process is based on stochastic models, the simulated occupancy changes every time, but the statistics of the simulated results are supposed to be the same. A simulation was conducted for three full working days. The total number of occupants in the building is shown in Fig. 5.

From Fig. 5, it can be seen occupants usually go to work at about 8:30 am, go for lunch at noon, and leave from work at about 17:30. During working hours, the number of occupants varies between 14 and 16, with some occupants being outside the building. The results show consistency with the input and similarity with the pre-determined schedule on a typical day, but when we zoom into the room level, the occupancy schedule becomes significantly different from the pre-determined one, as elaborated in Fig. 6.

It can be observed that the number of occupants in Office 2 fluctuates, with a maximum of six. For the meeting room, several meetings are held during a day, with the meeting durations being random, except for the scheduled afternoon meeting. In reality, each type of room has its own typical schedule, and the number of occupants fluctuates during the day, which cannot be reflected by the pre-determined occupancy schedules, but can be by the stochastic models.

Table 9
Settings of the rooms in the model.

	Room	No.	Time range	Times	Duration	Minimum	Maximum
1	Restroom	0	–	–	–	–	–
2	Researcher office	4	–	–	–	–	–
3	Supervisor office	1	–	–	–	–	–
4	Secretary office	1	–	–	–	–	–
5	Conference room	0	9:00–11:30 am	1 (random)	60 min	3	7
			2:00–3:00 pm	1 (fixed)	60 min	3	7
6	Kitchen	0	–	–	–	–	–
7	Researcher office	3	–	–	–	–	–
8	Researcher office	3	–	–	–	–	–
9	Manager office	1	–	–	–	–	–
10	Manager office	1	–	–	–	–	–
11	HR office	2	–	–	–	–	–
12	Corridor	0	–	–	–	–	–

Table 10
Settings of the occupants in the model.

ID	Type	Room	Own office		Other offices		Auxiliary room		Outside	
			Times	*P*	Times	*P*	Times	*P*	Times	*P*
A–D	Researcher	2	–	0.85	1	0.01	2	0.02	1	0.02
E	Supervisor	3	–	0.57	1	0.01	2	0.02	3	0.30
F	Secretary	4	–	0.63	1	0.03	2	0.02	2	0.10
G–I	Researcher	7	–	0.85	1	0.01	2	0.02	1	0.02
J–L	Researcher	8	–	0.85	1	0.01	2	0.02	1	0.02
M	Manager	9	–	0.72	1	0.01	2	0.02	2	0.15
N	Manager	10	–	0.72	1	0.01	2	0.02	2	0.15
O and P	HR staff	11	–	0.78	1	0.02	2	0.02	1	0.02

Occupant movement appears to be random on the surface, but actually it complies with some statistical discipline due to their occupations or habits. Therefore, it becomes important to capture this statistical discipline by the occupancy model, to ensure its applicability. Fig. 7 shows the simulation results of locations of two types of occupants.

Occupant A, who is a researcher, stays in his own room for more time than Occupant E who is a secretary. As inputs, a researcher spends 85% time during work, while a secretary only 63% and moves among rooms more frequently.

To explicitly show occupancy in the building at a particular time, some screen shots are presented. The number in the room is the number of occupants at that time and the letters below the room label denote the occupants who stay there at that time. Fig. 8 shows the occupancy at 12:00 pm on the first simulation day, where only a few occupants are in the building due to the lunch hour. Fig. 9 shows the results at 8:30 am, the arrival time for going to work. It was found that more than half of the occupants are already in the building. Fig. 10 shows the results at 14:30 pm when a scheduled meeting was to be held in the conference room labeled by 5. Four occupants participated in the meeting.

5.2. Co-simulation

Co-simulation is a simulation methodology that allows for individual components to be simulated by different simulation tools running simultaneously and to exchange information in a collaborative manner [29]. It exploits the modular structure of coupled problems in all stages of the simulation process beginning with the separate model setup and preprocessing for the individual subsystems in different simulation tools (which can be powerful simulators as well as simple C programs). During time integration,

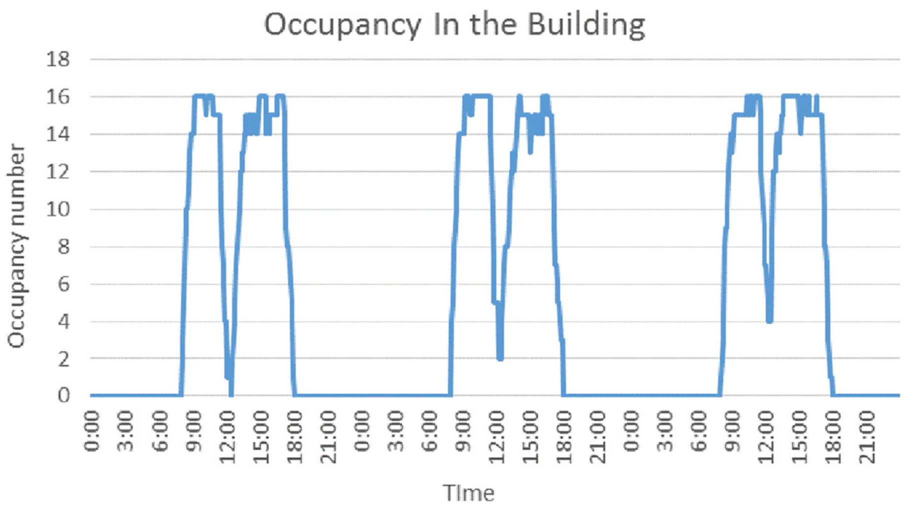

Fig. 5. Number of occupants in the building during 3 days.

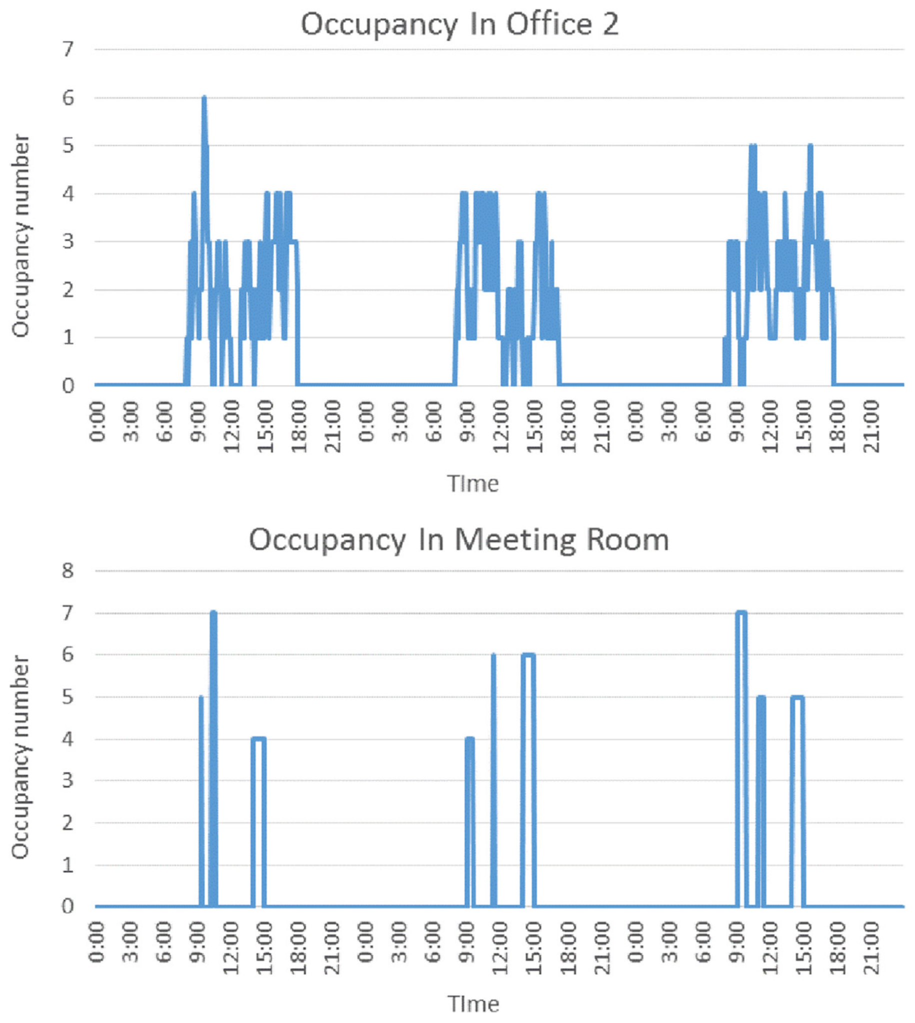

Fig. 6. Number of occupants in the Office 2 and meeting room during 3 days.

the simulation is again performed independently for all subsystems restricting the data exchange between subsystems to discrete communication points. In this way, modules developed by different programming languages or in different physical computers will be executed in an integrated manner.

The FMI (function mock-up interface) standard is a tool-independent standard to support both model exchange and co-simulation of dynamic models, using a combination of XML files, C-header files, and C-code in source or binary form [30,31]. FMI for co-simulation is an interface standard for coupling two or more simulation programs in a co-simulation environment. The data exchange between sub-systems is restricted to discrete communication points in time. In the time between two communication points, the sub-systems are solved independently from each other by their individual solver. A master algorithm controls the data exchange between sub-systems and the synchronization of all slave simulation programs (slaves). All information about the slaves, which is relevant for the communication in the co-simulation environment, is provided in a slave-specific XML file.

A simulation model or program which implements the FMI standard is called the functional mock-up unit (FMU). An FMU comes along with a small set of easy-to-use C-functions (FMI functions) whose input and return arguments are defined by the FMI standard. These C-functions can be provided in source and/or binary form. The FMI functions are called by a simulator to create one or more instances of the FMU. Fig. 11 shows a typical schematic diagram for co-simulation, where the master controls the FMU through a FMI, while the slave has its own model and solver to do simulation except when it communicates with the master [32].

To implement the FMI, an XML file is required to specify the information of variables in a module, e.g. properties like an input or output, constant or variable, reference, initial value, etc., which complies with an XML schema defined as the common model description. Besides, the functions declared in the header files should be defined and implemented before applying the FMU.

Unfortunately, all structure entities, like records or arrays, are flattened into a set of scalar values of type float, integer, etc. in current version of FMI [32]. Arrays are supposed to be defined in a fixed size, while the number of variables are uncertain in the occupancy module varies with the number of rooms, occupants or days to be simulated. The variables in the module cannot be enumerated in an XML file. An alternative maybe enumerating all variables in the XML file in the case where rooms, occupants and simulated days are already specified. However, when simulating a new case with a different number of rooms, occupants or simulated days, another XML file needs to be created, which is inconvenient and time-consuming. The FMI of the occupancy module will be fully implemented on condition that arrays with variable size are supported.

Co-simulation is a useful approach to running multiple simulation tools (can be multiple domains) simultaneously and

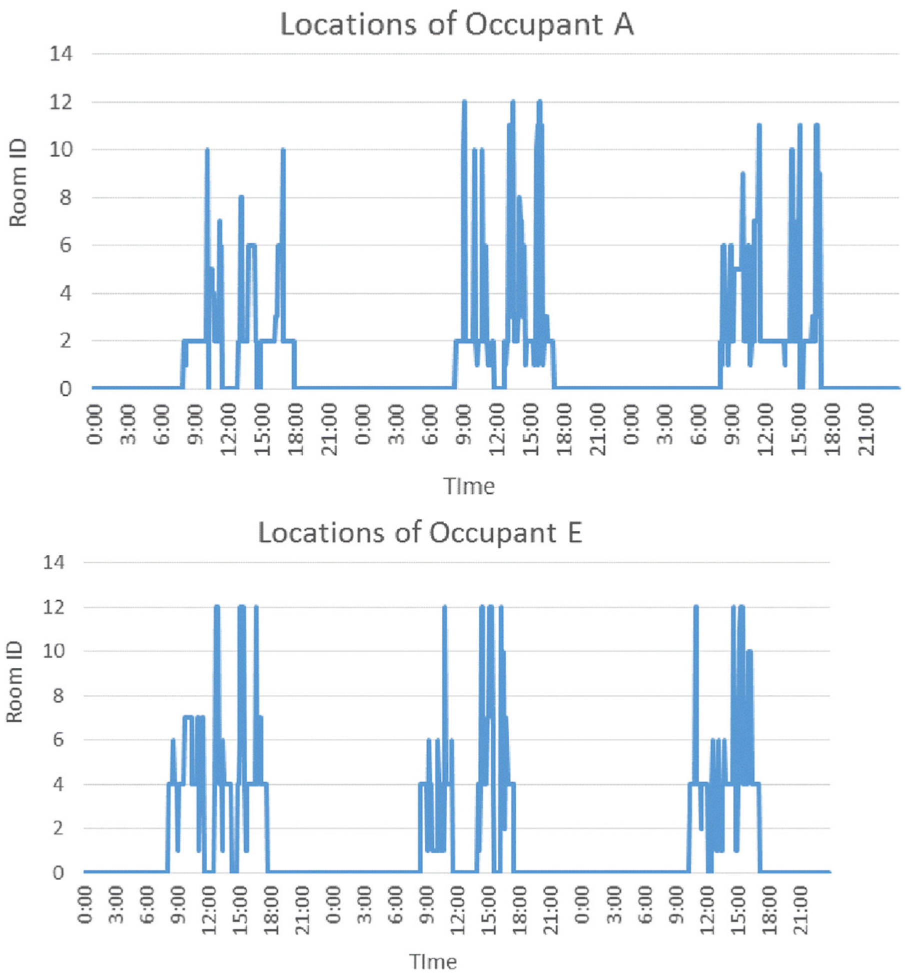

Fig. 7. Locations of Occupants A and E during 3 days.

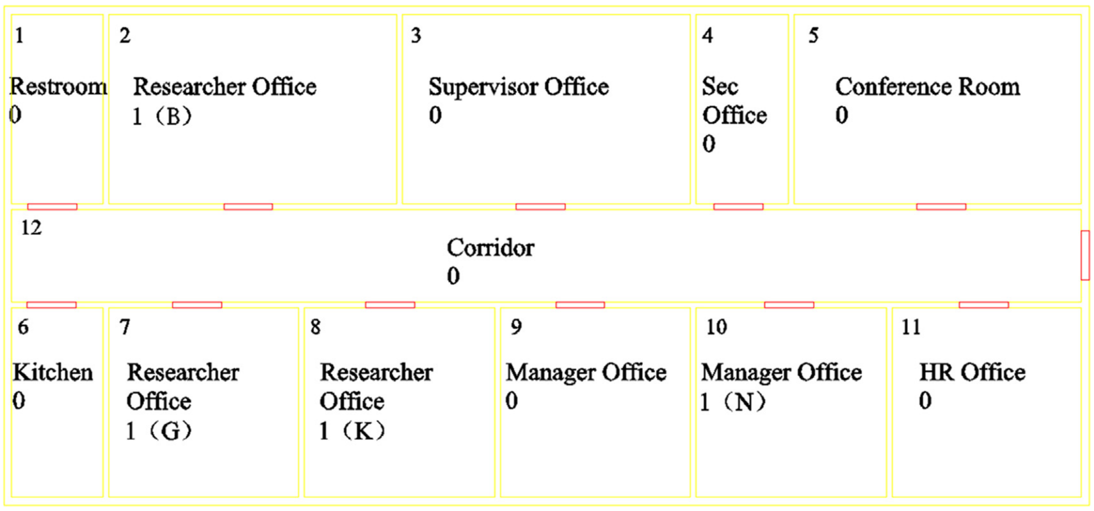

Fig. 8. Occupancy at 12:00 pm in the building on the first simulation day.

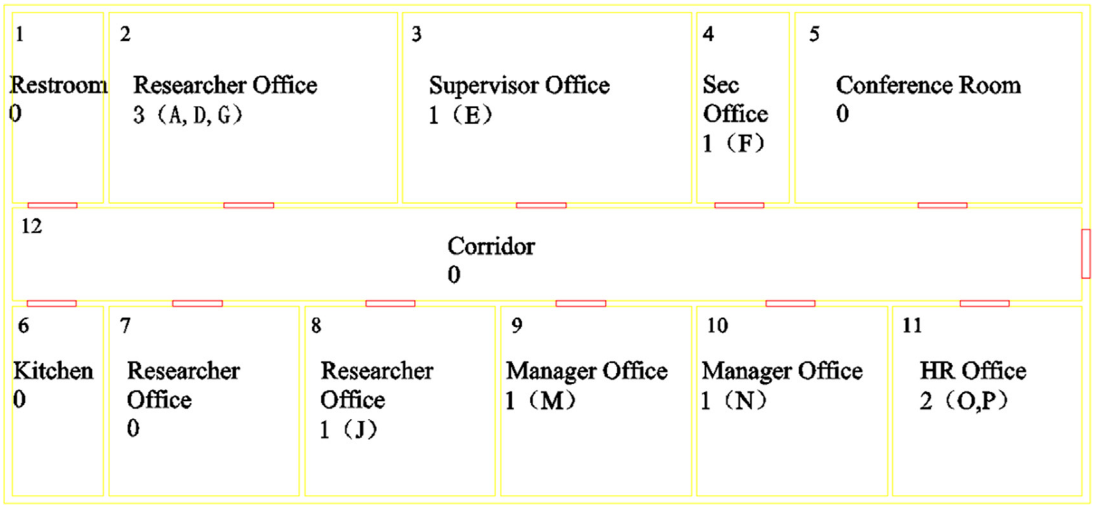

Fig. 9. Occupancy at 8:30 am in the building on the first simulation day.

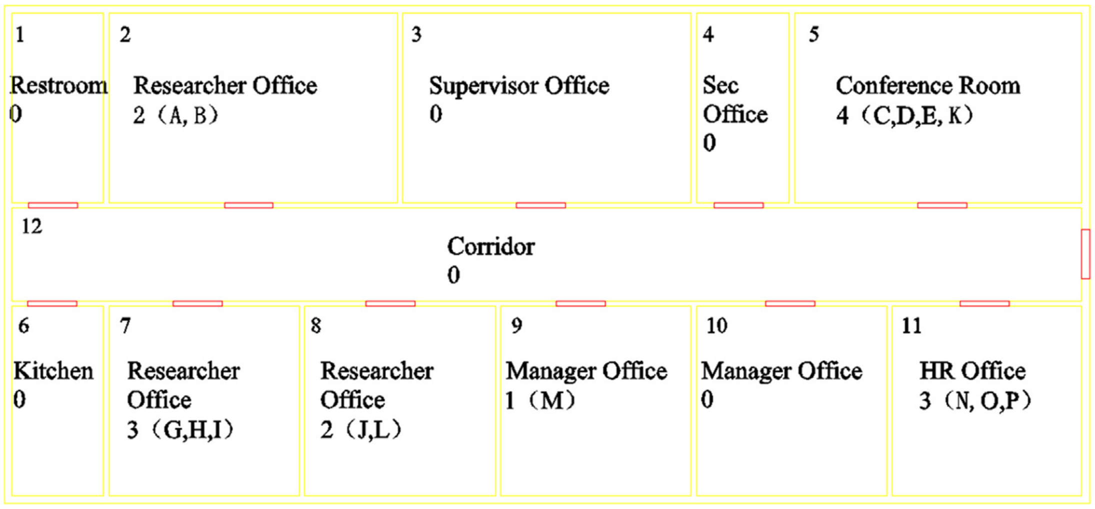

Fig. 10. Occupancy at 14:30 in the building on the first simulation day.

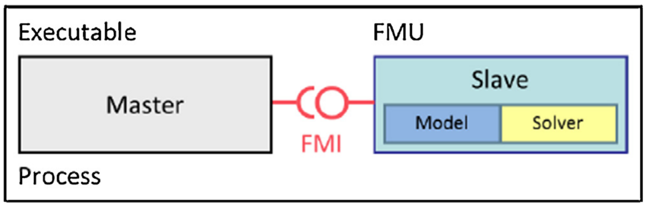

Fig. 11. Co-simulation with generated code on a single computer.

exchanging information in real-time. This allows for the ability to leverage the best features of different tools simultaneously to solve a hard problem or provide more detailed solutions.

6. Conclusion

The simulation of occupancy in buildings is fundamental for the simulation of the energy performance of buildings as well as the simulation of occupant behavior, as it provides the basic information of locations or presence of occupants. Current models try to address different levels of occupancy for diverse applications. For example, demand-controlled ventilation requires the number of occupants in a certain space, thus the space level number of occupants will suffice. A more detailed occupancy model may provide more concrete information, at the cost of more inputs to be specified and potentially longer computational time.

Four levels of occupancy modeling were identified to address various applications in buildings. Based on this comprehensive review, a software module was developed to include three representative occupancy models, which can be used to simulate various occupancy levels. The software architecture cohesively integrated occupancy state of a space, number of occupants in a space, and space location of individual occupants, a considerable advancement for the occupant behavior-building energy simulations community. Effort was made so that the software module was designed to be object oriented, flexible and extensible, allowing for the integration of new occupancy models and the ease of updates and maintenance.

The occupancy module can be used in multiple ways and for various audiences, including simulation users and software developers. The major part of the module was implemented in a DLL, which can be called by other simulation programs. The occupancy module can be used as a standalone program, to pre-calculate occupant schedules, generating schedule inputs which can be used with current energy modeling programs. To integrate with current simulation tools, the concept of co-simulation was introduced, which made it possible to be used with other tools. The FMI and FMU of the occupancy module will be fully implemented on condition that arrays with variable size are supported.

Acknowledgements

This work was sponsored by the United States Department of Energy (Contract No. DE-AC02-05CH11231) and the China Ministry of Housing and Urban–Rural Development and the Ministry of Science & Technology (Grant No. 2010DFA72740-02) under the U.S.-China Clean Energy Research Center for Building Energy Efficiency. It is also part of the research of Annex 66, Definition and Simulation of Occupant Behavior in Buildings, under the International Energy Agency Energy in Buildings and Communities Program.

References

[1] D.B. Crawley, L.K. Lawrie, F.C. Winkelmann, et al., EnergyPlus: creating a new-generation building energy simulation program, Energy Build. 33 (4) (2001) 319–331.
[2] D. Yan, J. Xia, W. Tang, et al., DeST—an integrated building simulation toolkit part I: fundamentals, Build. Simul. 1 (2) (2008) 95–110.
[3] V.L. Erickson, M.A. Carreira-Perpinan, A.E. Cerpa, OBSERVE: occupancy-based system for efficient reduction of HVAC energy, in: IEEE 10th International Conference on Information Processing in Sensor Networks (IPSN), 2011, pp. 258–269.
[4] T. Leephakpreeda, R. Thitipatanapong, T. Grittiyachot, et al., Occupancy-based control of indoor air ventilation: a theoretical and experimental study, Sci. Asia 27 (4) (2001) 279–284.
[5] X. Guo, D.K. Tiller, G.P. Henze, et al., The performance of occupancy-based lighting control systems: a review, Light. Res. Technol. 42 (4) (2010) 415–431.
[6] S. Pigg, M. Eilers, J. Reed, Behavioral aspects of lighting and occupancy sensors in private offices: a case study of a university office building, in: Proceedings of the 1996 ACEEE Summer Study on Energy Efficiency in Buildings, vol. 8, 1996, pp. 161–170.
[7] L.J. Lo, A. Novoselac, Localized air-conditioning with occupancy control in an open office, Energy Build. 42 (7) (2010) 1120–1128.
[8] G.Y. Yun, H. Kim, J.T. Kim, Effects of occupancy and lighting use patterns on lighting energy consumption, Energy Build. 46 (2012) 152–158.
[9] A. Al-Mumin, O. Khattab, G. Sridhar, Occupants' behavior and activity patterns influencing the energy consumption in the Kuwaiti residences, Energy Build. 35 (6) (2003) 549–559.
[10] G. Brager, G. Paliaga, R. De Dear, Operable Windows, Personal Control and Occupant Comfort, 2004.
[11] V. Pavlovas, Demand controlled ventilation: a case study for existing Swedish multifamily buildings, Energy Build. 36 (10) (2004) 1029–1034.
[12] W.J. Fisk, A.T. De Almeida, Sensor-based demand-controlled ventilation: a review, Energy Build. 29 (1) (1998) 35–45.
[13] Individual Prototype Building Models. Building Energy Codes Program, http://www.energycodes.gov/development/commercial/90.1_models
[14] C. Duarte, K. Van Den Wymelenberg, C. Rieger, Revealing Occupancy Patterns in Office Buildings Through the Use of Annual Occupancy Sensor Data, 2013.
[15] P. Bannister, Energy efficiency and the control and simulation of VAV systems, in: 2008 ACEEE Summer Study on Energy Efficiency in Buildings, 2008.
[16] T. Yu, Modeling occupancy behavior for energy efficiency and occupants comfort management in intelligent buildings, in: IEEE Ninth International Conference on Machine Learning and Applications (ICMLA), 2010, pp. 726–731.
[17] D. Wang, C. Federspiel, F. Rubinstein, Modeling occupancy in single person offices, Energy Build. 37 (2) (2005) 121–126.
[18] W. Chang, T. Hong, Statistical analysis and modeling of occupancy patterns in open-plan offices using measured lighting-switch data, Build. Simul. 6 (1) (2013) 23–32.
[19] R. Goldstein, A. Tessier, A. Khan, Space layout in occupant behavior simulation, in: Conference Proceedings: IBPSA-AIRAH Building Simulation Conference, 2011, pp. 1073–1080.
[20] J. Page, D. Robinson, N. Morel, et al., A generalised stochastic model for the simulation of occupant presence, Energy Build. 40 (2) (2008) 83–98.
[21] R. Goldstein, A. Tessier, A. Khan, Customizing the behavior of interacting occupants using personas, in: Proceedings of the National IBPSA-USA Conference, New York, USA, 2010.
[22] R. Goldstein, A. Tessier, A. Khan, Schedule-calibrated occupant behavior simulation, in: Proceedings of the 2010 Spring Simulation Multiconference, Society for Computer Simulation International, 2010, p. 180.
[23] I. Richardson, M. Thomson, D. Infield, A high-resolution domestic building occupancy model for energy demand simulations, Energy Build. 40 (8) (2008) 1560–1566.
[24] C. Wang, D. Yan, Y. Jiang, A novel approach for building occupancy simulation, Build. Simul. 4 (2) (2011) 149–167.
[25] K. Nassar, M. Elnahas, Occupant dynamics towards a new design performance measure, Archit. Sci. Rev. 50 (2) (2007) 100–105.
[26] C. Liao, P. Barooah, An integrated approach to occupancy modeling and estimation in commercial buildings, in: IEEE American Control Conference (ACC), 2010, pp. 3130–3135.
[27] V. Tabak, User Simulation of Space Utilisation, Eindhoven University Press, 2008.
[28] S. McConnell, Code Complete, O'Reilly Media, Inc., 2004.
[29] System Integration & Software Co-simulation. Flowmaster, http://www.flowmaster.com/flowmaster_integration.html (accessed 17.12.13).
[30] M. Wetter, Co-simulation of building energy and control systems with the building controls virtual test bed, J. Build. Perform. Simul. 4 (3) (2011) 185–203.
[31] T.S. Nouidui, M. Wetter, W. Zuo, Functional mock-up unit for co-simulation import in EnergyPlus, J. Build. Perform. Simul. (2013) 1–11.
[32] Function Mock-up Interface. Modelica Association Project, https://www.fmi-standard.org/ (accessed 17.12.13).