Schedule-Calibrated Occupant Behavior Simulation

Rhys Goldstein, Alex Tessier, Azam Khan
Autodesk Research
210 King St. East, Toronto, Ontario, Canada, M5A 1J7
Azam.Khan@autodesk.com

Keywords: Building performance simulation, energy demand prediction, occupant behavior, model calibration

Abstract

Building performance simulation promises to reduce the future impact of buildings on the environment by helping architects predict the energy demand associated with different design options. We present a new method for simulating occupant behavior in buildings, a key phase in the prediction of energy use. Our method first inputs the recorded activities of actual building occupants, then randomly generates fictional schedules with similar behavioral patterns. The main contribution of this work is a mathematical technique in which an arbitrary set of factors can be used to select plausible activity types, durations, and numbers of participants during a simulation. A prototype model was implemented to test the method, and results obtained to date suggest that the generated occupant schedules are believable when compared both qualitatively and quantitatively to real occupant schedules.

1 INTRODUCTION

It has been estimated that residential and commercial buildings account for 35 percent of global energy demand [1], as well as a substantial fraction of greenhouse gas emissions. At the same time, it is widely believed that design improvements could dramatically reduce the impact of buildings on the environment. Clarke identifies "ineffective decision-support" as the key factor impeding the possible realization of 50-75% reductions in the energy consumption of new buildings, and 30% reductions in that of existing buildings [2]. Our vision is that advances in building performance simulation will address the need for improved decision-support in building design, allowing architects to conveniently and accurately predict the energy use associated with different design options.

One of the more daunting aspects of building performance simulation is the number of different interacting subsystems that need to be modeled. These subsystems include the equipment in the building, the HVAC system, and the outdoor environment, among others [3]. Because a building's energy consumption patterns are largely dependent on the activities of its occupants, we chose to start our investigation of energy demand prediction by looking at occupant behavior. Quantifying this behavior is a prerequisite for predicting when a building's equipment is likely to be in use, and assessing the adequacy of its lighting conditions, air temperature, and air quality. As part of a long-term collaborative modeling project [4], our vision is to integrate realistic models of occupant behavior with those of other subsystems.

Existing building performance simulation tools typically use fixed schedules or relatively simple algorithms for modeling occupant behavior. In pursuit of more detailed and accurate predictions, researchers have proposed more sophisticated methods. It has been suggested that the trend towards flexible work hours will complicate occupant schedules and compound the need for these new methods [5]. Also, the emerging focus on sustainable buildings, incorporating passive air conditioning systems, will likely increase an occupant's influence over his or her surrounding indoor environment [6]. Simulations will need to capture these occupant-building interactions.

Here we present a novel occupant behavior simulation method. We describe our method as "schedule-calibrated", meaning that it first inputs the recorded activities of actual building occupants, then generates fictional schedules while striving to reproduce typical patterns of behavior. Several similar schedule-calibrated methods already exist. We contribute a mathematical technique that can be used to generate various activity "attributes". The specific attributes we focus on in this paper are the task performed, the number of participants, and the duration of each activity. Unlike pre-existing methods, ours allows each generated attribute to depend on an arbitrary set of "factors". Examples of factors include the time of day, the previously-performed task, and the time elapsed since each task was last performed. Any activity attribute may also be used as a factor. The generation of various attributes accommodates more detailed descriptions of human behavior, whereas the use of multiple factors promises to help reproduce behavioral patterns found in existing data.

Section 2 describes key concepts and reviews related work. Our proposed occupant behavior simulation method is explained in Section 3 using a simple example demonstrating the random selection of plausible tasks. In Section 4 we describe the actual prototype model. It was developed using the method of Section 3, but generates multiple attributes, uses a greater number factors, and discretizes those factors at higher resolutions. The prototype model was implemented, and simulation results are presented in Section 5. Finally, Section 6 discusses future work.

2 BACKGROUND

We use the phrase "occupant behavior simulation" to refer to a computer simulation that generates fictional occupant schedules. An "occupant schedule" is a description of the behavior of a building occupant over the course of a single day. Each of these schedules takes the form of a chronological sequence of consecutive activities, with an "activity" being a description of an occupant's behavior during a specific block of time. Each activity has several associated attributes. At very least, these attributes should include the "task", which identifies the type of activity. We expect each task to be selected from a pre-defined list of possible tasks. For a simulation of an office building, there would likely be one possible task representing desk work, another possible task for meetings, etc.

Several methods have been proposed to randomly generate plausible sequences of periods during which an occupant is either present or absent at a particular location in a building. While our interest lies in more detailed models of human behavior, these methods still satisfy our definition of "occupant behavior simulation" if the list of possible tasks is to include only "being present" and "being absent". Wang proposed that the durations of presence and absence be exponentially distributed [7], a convention which assumes that the remaining time to be spent in a location is independent of the time already spent there. Both Yamaguichi's method [8] and Page's method [9] also feature this "memoryless" property, but they differ from Wang's in that time is advanced by fixed time steps. Page introduced what we describe as a single influencing "factor"; specifically, the time of day. With this method, a simulation is calibrated using real schedules of presence and absence. If the real schedules tend to include a lunch break around noon, then the time of day factor allows that pattern of behavior to be reproduced.

Likely the most sophisticated occupant activity simulation developed to date is Tabak's User Simulation of Space Utilisation (USSU) System, described in his 2008 Ph.D thesis [10]. In USSU there are many different tasks, and occupants can interact via shared activities such as meetings and presentations. Unlike Page's method, USSU is not schedule-calibrated. Tabak instead conducted an extensive survey, using questionnaire results to calibrate his model.

The method developed for USSU [10] appears to have a few disadvantages. One possible concern is that a model calibrated using survey data might produce less realistic results than one calibrated using recorded schedules. The collection of the survey data itself is likely to be cumbersome (Tabak reported that 50 of 166 respondents failed to complete the questionnaire, presumably due to its length). That said, obtaining real occupant schedules for over 100 people would almost certainly be even more difficult. The drawback that concerns us most is the complexity of the USSU method. Based on the task performed, Tabak classifies activities as "Skeleton Activities" (eg. "give a presentation" or "do research"), "S-curve Intermediate Activities" (eg. "get a drink"), and "Probabilistic Intermediate Activities" (eg. "receive unexpected visitor"). Each type of activity is handled by a different algorithm, complicating the overall method.

3 PROPOSED METHOD

It seems plausible that the large amount of input data required by a schedule-calibrated method like Page's allows one to get away with relatively simple algorithms. Because the information contained in the real occupant schedules implicitly distinguishes one task from another, one can avoid explicit distinctions like those exhibited by USSU. But while Page's schedule-calibrated method is compelling for its simplicity, we decided at the outset of our work to avoid the memoryless property. Intuitively, an occupant's future behavior should be influenced in part by his or her past behavior. If one has taken a lunch break only three minutes ago, as opposed to three hours ago, one ought to be less likely to have a meal in the next three minutes. We therefore adopted the goal of developing a schedule-calibrated method like Page's, but with more detailed activities and an enhanced ability to reproduce observed patterns of behavior.

Recall that activities include attributes like the task, number of participants, and duration. In our proposed occupant behavior simulation method, attributes are generated one at a time using the same mathematical technique. Each attribute depends on an arbitrary set of factors, which for the time being must be chosen based on intuition. Intuitively, an occupant's next task should depend in part on the time of day, so like Page we might choose the time of day as a factor. Our method is novel in that it allows for the inclusion of other factors, the previous task being a notable example. Once one attribute is generated from a set of factors, that attribute can itself become a factor for the generation of another attribute. If one uses the time of day to generate the task, for example, one may then use the task as a factor that influences the activity duration.

Comparing our method to Page's and its predecessors, it is through the use of multiple factors that we expect an enhanced ability to reproduce behavioral patterns. Also, the generation of various activity attributes leads to a more detailed representation of occupant behavior. Striving to retain the simplicity of pre-existing schedule-calibrated methods, we generate each attribute using the same mathematical technique. This technique can be broken into four distinct phases: the population of a set of histograms using real occupant schedules; the smoothing of those histograms; the normalization of the smoothed histograms; and finally the extraction of attribute values. Each phase is explained in detail below.

PVT / TOD	A_0[…]				B_0[…][“off”]				B_0[…][“work”]				B_0[…][“meet”]				B_0[…][“eat”]			
	off	work	meet	eat	off	work	meet	eat	off	work	meet	eat	off	work	meet	eat	off	work	meet	eat
7 AM																				
8 AM	1								1											
9 AM	1								1											
10 AM		2												2						
11 AM			2								2									
12 PM		1																1		
1 PM		1		2								2						1		
2 PM																				
3 PM																				
4 PM		2	1			1					1			1						
5 PM		1				1														
6 PM																				

Figure 1. Histograms after the shaded bins were populated with the Table 1 data (empty bins represent values of 0).

3.1 Histogram Population

Throughout Section 3, an ongoing example will be used in which we generate a single activity attribute: the task. To simplify matters, we will have only four possible tasks, named “off”, “work”, “meet”, and “eat”. We will also use only two factors, the time of day (TOD) and the previous task (PVT).

Observe the two real occupant schedules in Table 1. Each row describes a separate activity. An activity begins at its associated time, and ends at the time when the subsequent activity begins. The “NPO” column lists the “number of participating occupants”. On Day 0, for example, the occupant had a meeting with four other occupants at 10:02 AM.

Table 1. Two real occupant schedules.

Day 0			Day 1		
Time	Task	NPO	Time	Task	NPO
8:45 AM	work	1	9:13 AM	work	1
10:02 AM	meet	5	10:41 AM	meet	2
11:17 AM	work	1	11:02 AM	work	1
12:10 PM	eat	3	1:16 PM	eat	1
1:01 PM	work	1	1:55 PM	work	1
5:47 PM	off	1	4:32 PM	meet	2
			4:51 PM	work	1
			4:59 PM	off	1

Our goal is to automatically generate tasks during a simulation, producing schedules that resemble those in the table. As mentioned earlier, the first step is to populate a set of histograms. Figure 1 shows five such histograms after they are populated using the data in Table 1. The histograms are all two-dimensional, as we have selected two factors for this example. There is one column for each of the four possible previous tasks, and the time of day factor is discretized such that each hour of the day has its own set of histogram bins.

Every activity in Table 1 contributes a value of 1 to both A_0 and one of the other four B_0 histograms. Each B_0 histogram is associated with a single “feature”, and in this specific example we happen to have one feature per possible task. The general mathematical technique we present allows one to choose any number of features, however, and each feature can be associated with any quantity. For the first activity of Day 0, the time (8:45 AM) fits into the “8 AM to 9 AM” slot and the previous task is assumed to be “off”. Thus $A_0[\text{“8 AM”}, \text{“off”}] = 1$, as shown just under the top-left corner cell of Figure 1. Because the task performed at 8:45 AM is “work”, we also add 1 to the corresponding bin of the “work”-specific feature histogram ($B_0[\text{“8 AM”}, \text{“off”}][\text{“}\mathit{work}\text{”}]$). It so happens that on both Day 0 and Day 1, the occupant transitions from the “work” task to the “meet” task between 10 AM and 11 AM. We therefore have $A_0[\text{“10 AM”}, \text{“work”}] = B_0[\text{“10 AM”}, \text{“work”}][\text{“meet”}] = 2$. On Day 1, the occupant transitioned away from the “work” task at 4:32 PM and 4:59 PM, giving us $A_0[\text{“4 PM”}, \text{“work”}] = 2$. Note that the 4:32 PM activity contributes to $B_0[\text{“4 PM”}, \text{“work”}][\text{“meet”}]$, whereas the 4:59 PM activity affects $B_0[\text{“4 PM”}, \text{“work”}][\text{“off”}]$.

3.2 Smoothing

The sparseness of Figure 1 is a problem, as we will eventually need to extract information from the bins that are currently empty. This problem will occur in practice, even if hundreds of real occupant schedules are used to populate the histograms, for the use of additional factors discretized at

TOD \ PVT	A_1[…]				B_1[…][“off”]				B_1[…][“work”]				B_1[…][“meet”]				B_1[…][“eat”]			
	off	work	meet	eat	off	work	meet	eat	off	work	meet	eat	off	work	meet	eat	off	work	meet	eat
7 AM	0.07								0.07											
8 AM	0.98	0.02	0.02	0.02					0.98	0.02	0.02	0.02								
9 AM	0.98	0.16	0.02	0.02					0.98	0.02	0.02	0.02		0.14						
10 AM	0.11	1.95	0.18	0.04					0.07		0.14		0.04	1.95	0.04	0.04				
11 AM	0.04	0.25	1.95	0.04					0.04	0.04	1.95	0.04		0.14				0.07		
12 PM	0.02	0.98	0.16	0.16							0.14	0.14					0.02	0.98	0.02	0.02
1 PM	0.06	0.99	0.06	1.95					0.04	0.01	0.04	1.95					0.02	0.98	0.02	0.00
2 PM		0.07		0.14								0.14						0.07		
3 PM		0.14	0.07			0.07					0.07			0.07						
4 PM	0.06	1.96	0.97	0.06	0.02	0.98	0.00	0.02	0.02	0.00	0.97	0.02	0.02	0.98	0.00	0.02				
5 PM	0.02	0.99	0.09	0.02	0.02	0.98	0.02	0.02			0.07			0.01						
6 PM		0.07				0.07														

Figure 2. Histograms after 1 smoothing iteration, with initially-populated bins shaded (“0.00” values are small but positive).

higher resolutions will increase the total number of bins. It is therefore necessary to “smooth” the data, propagating values across neighboring bins to reduce the sparseness.

Here we describe an iterative smoothing algorithm requiring one “smoothing parameter” $\alpha_{\langle factor\rangle}$ for each factor $\langle factor\rangle$. These smoothing parameters are all non-negative, and their sum is at most 1. The larger a smoothing parameter for a certain factor, the smoother the final data across neighboring bins along the axis associated with that factor.

The first step in each smoothing iteration i is to define a set of coefficients C_i, one for each histogram bin (identified by $\langle bin\rangle$) and each factor.

$$C_i[\langle bin\rangle][\langle factor\rangle] \;=\; \frac{\dfrac{\alpha_{\langle factor\rangle}}{n_{\langle factor\rangle}}\cdot \displaystyle\sum_{\langle factor\rangle} \alpha_{\langle factor\rangle}}{\displaystyle\sum_{\langle factor\rangle} \alpha_{\langle factor\rangle} \;+\; \sqrt{A_i[\langle bin\rangle]}}$$

If a bin has been heavily populated, then trusting its information, we lessen the effect of the smoothing by including $\sqrt{A_i[\langle bin\rangle]}$ in the denominator. The variable $n_{\langle factor\rangle}$ is the number of neighboring bins along the axis associated with $\langle factor\rangle$. For factors with continuous values like the time of day, the neighboring bins are the two adjacent bins (or the one adjacent bin if we are at the edge of a histogram). For discrete factors like tasks, all other bins along the axis are neighbors.

With c_i we record the fraction of each bin’s value that will be preserved through the smoothing iteration.

$$c_i[\langle bin\rangle] \;=\; 1 \;-\; \sum_{\langle factor\rangle} \left(n_{\langle factor\rangle}\cdot C_i[\langle bin\rangle][\langle factor\rangle]\right)$$

In a single iteration, convolutions are performed separately for each histogram using the same set of coefficients. Below, X_i represents any of the histograms for iteration i. We use $\langle bin^*\rangle$ to denote the neighboring bin located at a displacement $\langle offset\rangle$ along the axis associated with $\langle factor\rangle$.

$$\begin{aligned} X_{i+1}[\langle bin\rangle] \;=\; & c_i[\langle bin\rangle]\cdot X_i[\langle bin\rangle] \\ & + \sum_{\langle factor\rangle}\sum_{\langle offset\rangle} \left(C_i[\langle bin\rangle][\langle factor\rangle]\cdot X_i[\langle bin^*\rangle]\right) \end{aligned}$$

Continuing our simplified task generation example, we let $\alpha_{TOD} = 0.14$ and $\alpha_{PVT} = 0.06$. Figure 2 shows the histograms of Figure 1 after one iteration of smoothing. Serving as a demonstration of the algorithm, the following is a derivation of A_1[“4 PM”, “work”]. Note that its value is shown as 1.96 in the figure.

First, we calculate the smoothing coefficient associated with the time of day factor. Because this factor is continuous, there are two neighbors ($n_{TOD} = 2$).

$$\begin{aligned} & C_0[\text{“4 PM”}, \text{“work”}][\text{“TOD”}] \\ & \quad = \frac{\alpha_{\mathrm{TOD}}}{n_{\mathrm{TOD}}}\cdot\frac{\alpha_{\mathrm{TOD}} + \alpha_{\mathrm{PVT}}}{\alpha_{\mathrm{TOD}} + \alpha_{\mathrm{PVT}} + \sqrt{A_0[\text{“4 PM”}, \text{“work”}]}} \\ & \quad = \frac{0.14}{2}\cdot\frac{0.14 + 0.06}{0.14 + 0.06 + \sqrt{2}} \\ & \quad = 0.0086730\ldots \end{aligned}$$

The smoothing coefficient associated with the previous task is calculated in a similar fashion. In this case, because tasks are discrete and there are four in total, there are three neighbors ($n_{PVT} = 3$).

$$C_0[\text{“4 PM”}, \text{“work”}][\text{“PVT”}] \;=\; 0.0024780\ldots$$

TOD \ PVT	D[…][“off”]				D[…][“work”]				D[…][“meet”]				D[…][“eat”]			
	off	work	meet	eat	off	work	meet	eat	off	work	meet	eat	off	work	meet	eat
7 AM	0.00	0.00	0.00	0.00	1.00	0.81	0.98	0.98	0.00	0.19	0.02	0.02	0.00	0.00	0.00	0.00
8 AM	0.00	0.00	0.00	0.00	1.00	0.62	0.96	0.95	0.00	0.38	0.04	0.05	0.00	0.00	0.00	0.00
9 AM	0.00	0.00	0.00	0.00	0.98	0.15	0.81	0.72	0.02	0.85	0.19	0.28	0.00	0.00	0.00	0.00
10 AM	0.00	0.00	0.00	0.00	0.65	0.01	0.77	0.27	0.35	0.99	0.23	0.72	0.00	0.00	0.00	0.01
11 AM	0.00	**0.00**	0.00	0.00	0.79	**0.17**	0.99	0.81	0.13	**0.55**	0.01	0.13	0.08	**0.28**	0.00	0.06
12 PM	0.00	0.00	0.00	0.00	0.45	0.02	0.86	0.87	0.02	0.02	0.00	0.00	0.54	0.95	0.14	0.13
1 PM	0.00	0.00	0.00	0.00	0.64	0.06	0.71	0.99	0.00	0.00	0.00	0.00	0.35	0.94	0.29	0.01
2 PM	0.04	0.08	0.02	0.00	0.62	0.08	0.76	0.97	0.04	0.08	0.02	0.00	0.30	0.77	0.21	0.02
3 PM	0.30	0.47	0.04	0.17	0.38	0.02	0.90	0.66	0.28	0.46	0.04	0.16	0.04	0.05	0.01	0.01
4 PM	0.37	0.51	0.04	0.36	0.31	0.01	0.93	0.33	0.32	0.48	0.03	0.31	0.00	0.00	0.00	0.00
5 PM	0.74	0.90	0.25	0.74	0.15	0.00	0.73	0.15	0.11	0.09	0.02	0.11	0.00	0.00	0.00	0.00
6 PM	0.85	0.96	0.39	0.84	0.10	0.01	0.59	0.10	0.05	0.04	0.01	0.05	0.00	0.00	0.00	0.00

Figure 3. Normalized arrays, with initially-populated bins shaded (values shown in bold are referenced in the text).

The c_0 coefficient is obtained as follows.

$$\begin{aligned} c_0[\text{“4 PM”}, \text{“work”}] &= 1 \\ &\quad - n_{\text{TOD}} \cdot C_0[\text{“4 PM”}, \text{“work”}][\text{“TOD”}] \\ &\quad - n_{\text{PVT}} \cdot C_0[\text{“4 PM”}, \text{“work”}][\text{“PVT”}] \\ &= 0.97522\ldots \end{aligned}$$

Using A_1 in place of X_{i+1}, the convolution equation gives us the value shown in Figure 1.

$$\begin{aligned} &A_1[\text{“4 PM”}, \text{“work”}] \\ &= c_0[\text{“4 PM”}, \text{“work”}] \cdot A_0[\text{“4 PM”}, \text{“work”}] \\ &\quad + C_0[\text{“4 PM”}, \text{“work”}][\text{“TOD”}] \cdot A_0[\text{“3 PM”}, \text{“work”}] \\ &\quad + C_0[\text{“4 PM”}, \text{“work”}][\text{“TOD”}] \cdot A_0[\text{“5 PM”}, \text{“work”}] \\ &\quad + C_0[\text{“4 PM”}, \text{“work”}][\text{“PVT”}] \cdot A_0[\text{“4 PM”}, \text{“off”}] \\ &\quad + C_0[\text{“4 PM”}, \text{“work”}][\text{“PVT”}] \cdot A_0[\text{“4 PM”}, \text{“meet”}] \\ &\quad + C_0[\text{“4 PM”}, \text{“work”}][\text{“PVT”}] \cdot A_0[\text{“4 PM”}, \text{“eat”}] \\ &= 1.9616\ldots \end{aligned}$$

We terminate the smoothing processes after a pre-defined number of iterations n. If empty bins still remain, one can use a greater number of smoothing iterations, select lower resolutions on the continuous factors, or supply more input data.

3.3 Normalization and Extraction

After smoothing the histograms, the feature values B_n are normalized to yield a set of arrays collectively named D. This is the last step in the calibration process. In our ongoing example, normalization means dividing the B_n values by corresponding A_n values.

$$D[\langle bin \rangle][\langle feature \rangle] = \frac{B_n[\langle bin \rangle][\langle feature \rangle]}{A_n[\langle bin \rangle]}$$

Figure 3 shows the normalized arrays of the example after 9 iterations of smoothing. This information may be used during a simulation to generate plausible tasks, a process called extraction. Suppose that the current simulated time is 11:30 AM, and a simulated occupant has just completed a “work” activity. To select the next task, the simulation looks up the normalized feature values associated with these TOD and PVT factors.

$$\begin{aligned} D[\text{“11 AM”}, \text{“work”}][\text{“off”}] &= 0.00\ldots \\ D[\text{“11 AM”}, \text{“work”}][\text{“work”}] &= 0.17\ldots \\ D[\text{“11 AM”}, \text{“work”}][\text{“meet”}] &= 0.55\ldots \\ D[\text{“11 AM”}, \text{“work”}][\text{“eat”}] &= 0.28\ldots \end{aligned}$$

In this case there is roughly a 17% chance that the occupant will continue working, a 55% chance he/she will meet with other occupants, a 28% chance of taking a break for food, and very little chance he/she will leave for the day. The simulation selects the next task at random according to these probabilities.

Suppose that the occupant completes the “work” activity at 11:45 AM instead of 11:30 AM. In this case, if we choose, we may interpolate probabilities. The center of the “11 AM to 12 PM” slot is 11:30 AM, a quarter of an hour before the simulated time, and the center of the “12 PM to 1 PM” slot is 12:30 PM, three quarters of an hour after the simulated time. Using a linear interpolation, the probability that the occupant begins eating is $(3/4) \cdot 0.28 + (1/4) \cdot 0.95$, or 44.75%. The other probabilities would be interpolated in a similar fashion.

4 PROTOTYPE MODEL

Different models of occupant behavior may be defined using the method of Section 3, but with alternative sets of attributes, factors, and features. In a prototype model that we implemented to test the method, we generate for each activity its task, number of participating occupants (NPO), and duration. The same smoothing algorithm is applied in each case; our implementation uses multi-dimensional arrays to support an arbitrary number of factors. The histogram population, normalization, and extraction procedures differ somewhat between the three generated attributes, as explained below.

4.1 Task Generation

In our prototype, we generated tasks using three factors. Two of these, the time of day (TOD) and the previous task (PVT), were used in the simplified example of Section 3. The third factor is the "task suspension interval" (TSI).

The rationale for using TOD as a factor is that the timing of arrivals, departures, and lunch breaks is highly dependent on the time of day. We discretize TOD into 15-minute intervals, giving us a higher resolution than the 60-minute intervals used in the previous section. We use $\alpha_{TOD} = 0.23$.

The rationale for the PVT factor is that certain transitions between tasks are more likely to occur than others. We in fact chose $\alpha_{PVT} = 0$, which prevents a transition from occurring in the simulated schedules if it is not found in the real schedules. With $\alpha_{PVT} = 0$, one would not expect two washroom breaks to be generated back-to-back.

The TSI measures the time elapsed since a task was last performed. Its inclusion as a factor helps reproduce intervals between tasks. This is useful to spread out washroom breaks, or coffee/lunch breaks, for example. We adopted a logarithmic discretization, with separate bins for task suspension intervals of 0 to 15 minutes, 15 to $15\cdot\sqrt{2}$ minutes, $15\cdot\sqrt{2}$ to 30 minutes, 30 to $30\cdot\sqrt{2}$ minutes, etc. We selected $\alpha_{TSI} = 0.02$.

Whereas the TOD and PVT factors are scalar values, the TSI factor requires a separate time interval for each possible task. This complicates the mathematical procedure demonstrated in Section 3. When populating a histogram with a single activity, or when extracting a set of probabilities, multiple TSI values must be used to reference histogram bins.

4.2 NPO Generation

To generate the numbers of participating occupants for each activity, we again use the TOD factor. Because we decided to generate an activity's NPO after generating its task, we are able to use the current task (TSK) as the other factor.

We believe TOD might well influence the NPO, as an occupant taking a break is more likely to have company around noon than in mid-afternoon. We again discretize the TOD at 15-minute intervals, but now use $\alpha_{TOD} = 0.25$.

It is obvious that the NPO should depend on the TSK, as a task representing meetings is more likely to be collaborative than, for example, a washroom break. A α_{TSK} value of 0 ensures that washroom breaks are always treated as individual activities.

When generating the task in Section 3, we required one feature per possible result. When generating the NPO, there are far more possible results. Here we use exactly two features, one called "sums" and another called "square_sums". For each activity in the real schedules, we first subtract $1/2$ from the number of participants npo. We add $(npo - 1/2)$ to the correct bin in the "sums" feature histogram, and $(npo - 1/2)^2$ to the same bin in the "square_sums" histogram. After n iterations of smoothing, we have for each bin $\langle bin\rangle$ the values $A_n[\langle bin\rangle]$, $B_n[\langle bin\rangle][\text{“}sums\text{”}]$, and $B_n[\langle bin\rangle][\text{“}square\_sums\text{”}]$. The normalization equations are as follows.

$$D[\langle bin\rangle][\text{“}mean\text{”}] = \frac{B_n[\langle bin\rangle][\text{“}sums\text{”}]}{A_n[\langle bin\rangle]}$$

$$D[\langle bin\rangle][\text{“}variance\text{”}] = \frac{B_n[\langle bin\rangle][\text{“}square\_sums\text{”}]}{A_n[\langle bin\rangle]} - D[\langle bin\rangle][\text{“}mean\text{”}]^2$$

During a simulation, with the time of day and the current task known, appropriate mean and variance values can be interpolated from D. We formulate a gamma distribution with these properties, randomly select a positive value from this distribution, add 1 to that value, then convert that real number to a positive integer by rounding down. The result is the NPO.

4.3 Duration Generation

We decided that the duration of an activity should depend on the time of day (TOD), the task (TSK), and the number of participating occupants (NPO).

The rationale for the TOD factor is that, for example, an occupant is less likely to work for several hours continuously if it is nearly time to leave the office. We again select 15-minute intervals, and use $\alpha_{TOD} = 0.23$.

It is obvious that an activity's duration is highly dependent on the task; in fact we insist that $\alpha_{TSK} = 0$. A positive α_{TSK} leads to absurdly long washroom breaks, as their durations become influenced by those of other types of activities.

The NPO factor is used because we imagine that an occupant may take a longer lunch break if he/she has company. We use separate bins for activities with 1 participant, 2 or 3 participants, 4 to 7, 8 to 15, 16 to 31, 32 to 63, and 64 or more participants. For smoothing, $\alpha_{NPO} = 0.02$.

Duration generation is similar to NPO generation. We again use two features, adding recorded durations to one feature and their squares to the other. This allows us to randomly generate durations using gamma distributions. As a duration is a continuous quantity, we do not need to offset or round off values as we did for the NPO.

5 RESULTS

Ideally, the fictional schedules generated by a our occupant behavior simulation would be indistinguishable from the real schedules used to calibrate the model. There is no single "best" metric to determine how well the two sets of schedules resemble one another. Here we present a brief qualitative analysis of our results, followed by a few statistics. In each case, the model was calibrated using the same 27 real schedules. These schedules were recorded manually during weekdays by an Autodesk Research employee, referred to in this section as "the real occupant". On some days, each activity was entered into a spreadsheet shortly after its completion. In other cases the occupant recorded all activities at the end of the day, aided by a webcam and motion detection software.

Observe the generated schedule in Table 2. The schedule appears plausible in several regards: the arrival and departure times; the fact that the occupant spends most of the day doing desk work; the 6-minute "desk meeting" in the morning; and the fact that the washroom breaks are spread out and range from 1 to 4 minutes in duration. The 13-person lunch break outside the building happens to be consistent with a few of the 27 real schedules.

Table 2. A generated occupant schedule.

Time	Task	NPO
9:57 AM	work@desk	1
10:19 AM	meet@desk	2
10:24 AM	work@desk	1
11:37 AM	break@washroom	1
11:41 AM	work@desk	1
12:01 PM	break@sharedroom	1
12:12 PM	break@washroom	1
12:13 PM	work@desk	1
12:28 PM	break@outside	13
1:31 PM	break@washroom	1
1:34 PM	work@desk	1
2:42 PM	break@washroom	1
2:44 PM	work@desk	1
4:08 PM	break@sharedroom	1
4:13 PM	work@desk	1
6:13 PM	off@outside	1

The real occupant never left the building and returned twice in a single day. This pattern of behavior appeared to be reflected in the generated schedules, presumably due to the TSI factor that measures the time elapsed since a task was last performed. But because breaks outside the building ("break@outside") were classified as separate tasks from breaks on site ("break@sharedroom"), the TSI factor did nothing to prevent the outside break in Table 2 from being preceded by the break at 12:01 PM. By combining "break@outside" and "break@sharedroom" into a single task, a modeler could discourage the generation of multiple breaks around noon. But on the other hand, one might then lose the tendency for a 13-person "team lunch" to take place at a restaurant, or the trend that 5-minute breaks around 4:08 PM tend to occur on site. The more general point is that different classification schemes will reproduce different behavioral patterns, but no single classification scheme will be ideal.

Figure 4 shows profiled probabilities that an occupant can be found working at their desk. The jagged line was produced by counting, for each minute of the day, the number of times the real occupant was recorded performing desk work, then dividing by the total number of recorded schedules (27). The smoother line was calculated in the same fashion, but using 10000 generated schedules. The real and simulated profiles follow roughly the same path, though evidently the simulation overestimates the probability of desk work around lunch and underestimates it before and after.

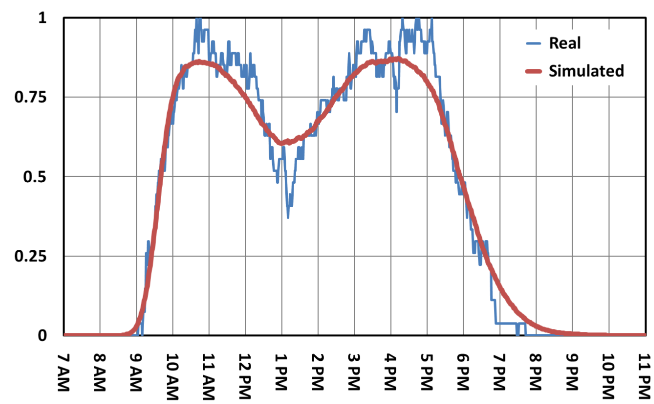

Figure 4. Desk work probability profiles.

Despite the differences between the profiles at certain hours of the day, the real and simulated time-averaged probabilities of desk work were in fairly close agreement. On average, the real occupant worked at their desk only 3.4 minutes longer per day than the simulated occupant (411.6 minutes total compared with 408.2).

While desk work is typically performed several times per day, the occurrence of certain other tasks is best measured over several days. The real occupant took a break outside on 14 of 27 days, and so the desired probability that the "break@outside" task occurs in a day is about 51.9%. From 10000 generated schedules, the result was roughly 53.6%. Given that we are trying to predict human behavior, the error of 1.7% seems tolerable.

6 FUTURE WORK

Recall that Tabak's USSU System [10] allowed simulated occupants to interact with one another, sharing activities such as meetings and presentations. Although we do quantify the number of participants for each activity, we have yet to present a means to utilize this attribute for occupant interaction. For example, suppose our method is used to simulate a building with 10 occupants over the course of a single day. If at some point a 5-person activity is generated for one of those occupants, then the same activity should occur at the same time in the schedules of 4 of the remaining occupants. So long as the 10 schedules are generated independently, however, this is extremely unlikely. Adding interactions between occupants, such that simultaneously-generated schedules are inter-dependent, remains important future work.

Discussing the schedule-calibrated method of [9], Page points out that software users will find it impractical to supply the large amounts of input data necessary to yield realistic simulated behavior. The problem applies to our own method as well, and is exacerbated by the possibility that users will want to populate their simulated buildings with different types of occupants. Designing a building for software company, for example, an architect may wish to perform simulations with different numbers of junior programmers, senior programmers, managers, and sales and support staff, each with their own behavioral patterns. We hope to combine our method with personas, descriptions of fictional individuals, to allow occupant behavior to be customized using only modest amounts of additional information.

Once we have developed a customizable model of the behavior of interacting occupants, we will need to combine it with models of other subsystems in an effort to predict building performance. At very least, the occupant behavior model should influence models of building equipment. If a simulated occupant begins performing desk work, for example, his/her simulated computer should respond and draw additional power. It is important to note that interactions can occur in the opposite direction as well, with other building subsystems influencing human behavior. If an HVAC system produces an intolerable temperature increase in a working area, an occupant might respond and move to a different location. Alternatively, he/she may open a window, impacting the HVAC system and potentially the actions of other occupants. In some cases it may be desirable to allow "exceptional behavior", like opening a window or vacating an uncomfortable area, to take precedence over the activities generated by our schedule-calibrated method.

7 CONCLUSION

A number of methods have previously been developed to simulate the behavior of building occupants based on the recorded schedules of real occupants. The schedule-calibrated method we have proposed and demonstrated is notable for its flexibility; one can determine the level of detail with which occupant behavior is modeled by selecting various activity attributes, and one can alter the behavioral patterns that get reproduced in a simulation by selecting different sets of factors. Tested with our own chosen attributes and factors, the method yielded plausible fictional schedules with acceptable statistical accuracy. Future work includes the modeling of interactions between occupants, the customization of occupant behavior using personas, and the integration of occupant models with those of other building subsystems in an effort to predict energy demand.

REFERENCES

[1] McKinsey Global Institute. *Curbing Global Energy Demand Growth: The Energy Productivity Opportunity*, 2007.

[2] Joe A. Clarke. *Energy Simulation in Building Design*. Butterworth-Heinemann, second edition, 2001.

[3] Jan Hensen, Ery Djunaedy, Marija Radošević, and Azzedine Yahiaoui. Building Performance Simulation for Better Design: Some Issues and Solutions. In *Proceedings of the 21st Conference on Passive and Low Energy Architecture (PLEA)*, pages 1185–1190, Eindhoven, Netherlands, 2004.

[4] Autodesk Research, CIMS Lab, and Faro Technologies. *Digital 210 King Project*. www.digital210king.org.

[5] Gerhard Zimmerman. Modeling and Simulation of Individual User Behavior for Building Performance Predictions. In *Proceedings of the Summer Computer Simulation Conference (SCSC)*, pages 913–920, San Diego, California, 2007.

[6] Pieter-Jan Hoes, Jan Henson, M.G.L.C. Loomans, B. de Vries, and D. Bourgeois. User behavior in whole building simulation. *Energy and Buildings*, 41(3):295–302, 2009.

[7] Danni Wang, Clifford C. Federspiel, and Francis Rubinstein. Modeling occupancy in single person offices. *Energy and Buildings*, 37(2):121–126, 2005.

[8] Yohei Yamaguchi, Yoshiyuki Shimoda, and Minoru Mizuno. Development of District Energy System Simulation Model Based on Detailed Energy Demand Model. In *Proceedings of the Eigth International IBPSA Conference*, pages 1443–1450, Eindhoven, Netherlands, 2003.

[9] Jessen Page. *Simulating Occupant Presence and Behaviour in Buildings*. PhD thesis, École Polytechnique Fédérale de Lausanne, Switzerland, 2007.

[10] Vincent Tabak. *User Simulation of Space Utilisation: System for Office Building Usage Simulation*. PhD thesis, Eindhoven University of Technology, Netherlands, 2008.