A MARKOV-CHAIN ACTIVITY-BASED MODEL FOR PEDESTRIANS IN OFFICE BUILDINGS

Sanmay Shelat[1], Winnie Daamen[1], Bjorn Kaag[2], Dorine Duives[1], Serge Hoogendoorn[1]
[1]Department of Transport and Planning, Delft University of Technology
Stevinweg 1, Delft, The Netherlands
s.shelat@tudelft.nl
[2]Argosonic, The Netherlands

Abstract **-** As the number of people working in office buildings increases, there is an urgent need to improve building services, such as lighting and temperature control, within these buildings to increase energy efficiency and well-being of occupants. A pedestrian behaviour model that simulates office occupants' movements and locations can provide the high spatial and temporal resolution data required for the testing, evaluation, and optimization of these control systems. However, since most studies in pedestrian research focus on modelling specific actions at the operational level or target situations where movement schedules do not have to modelled, a pedestrian behaviour model that can simulate complex situations over long time periods is missing. Therefore, this paper proposes a tactical level model to generate occupant movement patterns in office buildings. The Markov-chain activity-based model proposed here is data parsimonious, flexible in accepting different levels of information, and can produce high resolution output. The mathematical properties of the methodology are analyzed to understand their impact on the final results. Finally, the tactical level pedestrian behaviour model is face validated using a case study of an imaginary office with a simple layout.

Keywords**:** pedestrians; tactical behaviour level; offices; buildings; activity-based; Markov-chain

1. Introduction

Around the world, as more people work in office buildings, these buildings consume a large proportion of resources for building services, such as lighting and temperature control. Therefore, these services need to be researched and evaluated to optimize their performance, for which occupant behaviour data, such as their spatio-temporal distribution within the building, is required. A pedestrian behaviour model that simulates the movements of office occupants can generate this data for various situations without expensive implementation of test sensors in the real-world and minimal privacy concerns. However, most pedestrian studies focus on the operational behaviour level [1] and those that do model a broader picture either assume a movement schedule or model situations where the sequence of movements is already known. To obtain pedestrian locations over long periods of time in complex situations, the tactical behaviour level, where pedestrian itineraries are constructed, must be modelled alongside the operational level. Hence, the next step in pedestrian research is to develop a model that integrates the different behaviour levels [2]. For this, the tactical behaviour level which schedules movements deserves greater attention than it has received.

The few approaches in literature for scheduling pedestrian movements can be divided into (i) activity-based, (ii) location-based, and (iii) random-access models. In the first category, movements take place because pedestrians have to perform activities at different locations. Activity episodes are generated and scheduled for pedestrians. A location choice model is used to assign each activity episode in the itineraries to a location to form movement schedules. Tabak [3] presented the most detailed model of pedestrian behaviour in offices but, like other models in this category, it has been critiqued for its complexity and extensive data requirements. In the next two categories, movement occurs for movement's sake and pedestrians are directly assigned a sequence of locations to visit, making these models less intuitive than activity-based ones. Location-based models generate individual movements using stochastic models with the aim of reproducing location-based statistics, such as occupancy or transition sequences. A common method in location-based models is to divide the area modelled into zones and use them as states of Markov-chains that are assigned to every pedestrian. The Markov-chains are then simulated in discrete time steps and a movement is generated whenever a pedestrian transitions to another zone [4, 5]. These models are much less complex and data intensive than the previous category. However, unlike activity-based models, the running time and complexity of location-based models depends on the resolution of the zones and the size of the area modelled. Finally, random-access models use random walks that do not consider locations or activities but depend on the overall spatial configuration to simulate movements [6]. However, they are generally not applicable for situations that are process driven or buildings that are highly programmed and, therefore, not useful for scheduling movements in office buildings.

This paper proposes a tactical level model of pedestrian behaviour in office buildings that is data parsimonious, flexible in accepting different amounts of information, and can produce high resolution output; thus combining the advantages of activity- and location-based approaches whilst overcoming their limitations. The focus here is on the scheduling of movements while other decisions at this level, such as route and activity location choice, are simplified with assumptions that office occupants are familiar with the building and indifferent to activity locations, which is reasonable given the types of activities to be performed (e.g. getting coffee) [7]. Furthermore, unlike previous studies [5, 8, 9] the mathematical properties of the stochastic process used in the model are analysed to understand their effect on the final results. The methodology and its analysis are presented in section 2. In order to face validate the model, a simple case study of an imaginary office is carried out in section 3, followed by the conclusions in section 4 which note model limitations and relevant future work.

2. Methodology

To combine the advantages of the activity- and location-based approaches, the Markov-chain location-based model (henceforth, the Wang model) proposed in [5] is adopted in an activity-based framework. The Wang model is similar to many models in its category in that it divides the building into zones which are used as states in the Markov-chains assigned to each occupant, and these Markov-chains are simulated to generate movements. However, recognizing that defining each transition matrix element in every agent's Markov-chain is data-intensive, the authors of the Wang model propose a novel method that derives these probabilities from profiles consisting of only two variable sets per agent; thus, considerably reducing the number of inputs required to model multi-zone, multi-occupant scenarios. Other studies [8, 9] have used the Wang model to develop occupancy simulation software because of its simplicity and data parsimony.

However, certain limitations prevent the Wang model from being used to simulate pedestrian movements. Being a location-based model, it is not robust against building size and cannot use high spatial resolutions. Furthermore, all movements between zones are assumed to take place in a single time step which lowers the temporal resolution to maintain feasibility of the assumption (a 5 minute time step is used in [5]). These limitations are resolved by adopting the Wang model in an activity-based framework (section 2.2) and adding complementary models (section 2.4). Additionally, their novel method of generating transition matrices is improved by linearizing the optimization problem (section 2.2) and analysing its impact on the final results through simulation (section 2.3).

2.1. Activity Classification

Since activities have different properties, they cannot all be generated and scheduled with the same methodology. Thus, to make the activity-based framework possible, a classification of activities is created such that each category contains activities with similar properties whilst all categories together allow representing the full range of activities occurring in various situations [3, 10]. These categories are further grouped according to their association with a time-of-day (time-dependent or time-independent) and these groups each have their own scheduler.

Time-dependent activities include *planned* and *time-window* activities, which, respectively, have a fixed starting time or a fixed time window within which they may be performed. Planned activities (e.g., meetings) have priority over all other activity types, whereas time-window activities (e.g., arrivals, lunch) have a higher priority than time-independent activities but may be skipped if they cannot be executed in their time window due to planned activities. These activities are modelled using an *Event Scheduler*. Time-independent activities include recurrent and continuous activities. Recurrent activities are related to occupants' physiological processes [3] (e.g., taking breaks, getting a drink, visiting the restroom) which may be assumed to have a regular recurrence time. It is assumed that these activities are not undertaken in the middle of time-dependent activities, such as meetings. Continuous activities (e.g., working at one's desk), which have the lowest priority, are performed when no other activity is being performed. They do not have a fixed starting time, duration, or period and are the default activity to which occupants return when they are unable to execute other activities. The *Markov-chain Scheduler*, which is a modification of the Wang model, is used to simulate time-independent activities.

2.2. Markov-Chain Scheduler

The methodology used to generate and schedule time-independent activities is based on first-order, discrete time, finite-space ergodic Markov-chains. Instead of the zones, to adopt the Wang model in an activity-based framework, time-independent activities are used as the states of the Markov-chains. Eq. 1 describes the memoryless characteristic of Markov chains with states S indicating that the state of the chain, X, at time t, is only dependent on the state in the previous time-step. Combining p_{ij}'s for all i and j into matrix form creates a transition matrix P, where each element (p_{ij}) indicates the probability of going from the row state (i) to the column state (j). Eq. 2 gives an example of an n-state transition matrix.

$$P\{X_t = j|X_0 = m, X_1 = k, \dots, X_{t-1} = i\} = P\{X_t = j|X_{t-1} = i\} = p_{ij},$$
$$\forall i, j, \dots, m, n \in S. \tag{1}$$

$$P = \begin{bmatrix} p_{11} & \cdots & p_{1n} \\ \vdots & \ddots & \vdots \\ p_{n1} & \cdots & p_{nn} \end{bmatrix} s.t. \textstyle\sum_{j=1}^{n} p_{ij} = 1;\ \forall i, j\ p_{ij} \geq 0. \tag{2}$$

The model uses two inputs to generate the transition matrix for each occupant. The first input is the average continuous duration of time (also called expected sojourn time) the agent spends in a state, that is, a time-independent activity. For large values of t, (i.e., $t \to \infty$) the expected sojourn time (τ_i) in a state converges to a limiting value which can be used to derive the diagonal elements of the transition matrix as shown in Eq. 3. The second input is the proportion of time spent in a state, that is performing a time-independent activity. For $t \to \infty$, this value – the probability of being in a state, which describes, in essence, the overall percent of time a Markov-chain spends in a given state – also converges to a limit. This probability distribution over all the states of a Markov-chain is called the stationary probability distribution (π) (Eq. 4). By definition, the stationary probability distribution can be calculated as the left eigenvector (normalized to 1) of the transition matrix (Eq. 5). For recurrent activities, such as getting a coffee, it is more convenient to use the average time between two episodes of the activity, that is the away time of a state (α_i). This variable is also observed to tend to a limit which can be used to derive the stationary probability values of those states (Eq. 6).

$$E(\tau_i) = \frac{1}{1 - p_{ii}} \Rightarrow p_{ii} = 1 - \frac{1}{E(\tau_i)}. \tag{3}$$

$$\pi = [\pi_1, \pi_2, \dots, \pi_n]\ s.t. \textstyle\sum_{i=1}^{n} \pi_i = 1;\ \forall i\ \pi_i \geq 0. \tag{4}$$

$$\pi = \pi \cdot P. \tag{5}$$

$$E(\alpha_i) = E(\tau_i) \cdot \left[\frac{1}{\pi_i} - 1\right] \Rightarrow \pi_i = \frac{E(\tau_i)}{E(\tau_i)+E(\alpha_i)}. \tag{6}$$

For each occupant, a transition matrix is derived by setting up a constrained linear least-squares problem. Linearization increases the speed of the model considerably, thus, reducing the model's running time. The system of linear equations is derived from the following conditions: (i) the relation between the stationary probability distribution and the transition matrix (Eq. 5); (ii) the fact that the sum of columns of a row in the transition matrix sum to 1 (Eq. 2); (iii) the known values of the diagonal elements of the transition matrix (Eq. 3); and (iv) the constraint that all transition elements are non-negative values (Eq. 2). Thus, for an n-state Markov chain, the system of equations consists of $2n$ equations, formed by the first two conditions, which are solved for $n(n - 1)$ variables that represent the elements of the $n \times n$ transition matrix (given that the diagonal elements are known from the third condition). Eq. 7 shows the system of linear equations used for the constrained linear least-squares problem and Eq. 8 shows the setup of the problem.

In order to generate the transition matrix, MATLAB's constrained linear least-squares solver *lsqlin* is used to solve Eq. 8 for x – the array of unknown transition matrix elements. Similar to [5], the problem proposed in Eq. 8 is under-determined, and thus, it has either no solutions or an infinite number of solutions. Since a unique solution is not already defined, the linear problem can accept increasing amounts of information through additional linear constraints of the form $Cx \leq d$ or upper and lower bounds for x in the constrained least-squares problem. This additional information can be in the form of complementary observations, such as the tendency to perform one activity after another. Incorrect solutions are discarded by setting a tolerance value of 10^{-10} on the objective function value of *lsqlin*, which should ideally be zero. Furthermore, since this method uses ergodic Markov-chains, a check is made to detect and remove states with zero stationary probability (i.e., activities that are never performed by an occupant) when generating a transition matrix. After the generation, null rows and columns corresponding to those states are added to maintain consistency with other transition matrices.

Once the transition matrix is generated, a simulation of the Markov chain is carried out by starting from a random initial state (activity) and then choosing the next state at every time step using the transition probabilities from the current state. This simulation returns a sequence of time-independent activities for the occupant. Thus, doing both, generating activity episodes with durations and scheduling them. The next section analyses how the under-determined nature of the problem impacts the final results.

2.3. Transition Matrix Generation Analysis

Since the problem setup in Eq. 8 is under-determined, the same inputs will result in more than one transition matrices (if any). This could mean that if the solver resulted in a different transition matrix, the end results could be quite different. However, while the problem setup in the previous section is, strictly speaking, only true for expected values at $t \to \infty$, the Markov-chains are simulated for a maximum time period of a day, which is approximately 500 time steps if each time step is a minute (500 minutes ~ 8 hours ~ 1 working day). Hence, the limited time simulations add stochasticity to the results, which could mean that the existence of different transition matrix solutions may not, ultimately (i.e., post-simulation), make a difference. To check this, a three-state situation is considered. Four Markov-chains are obtained from the same inputs and simulated 1000 times for 500 time steps. The transition probability observed from the simulation (i.e., the simulated transition probabilities) between all the states are recorded for each run and plotted as an histogram with different colours representing the four Markov-chains (Fig. 1). Additionally, to understand the impact of the input values on the simulated transition probabilities the sojourn time values are varied – the bottom figure of Fig. 1 has 10 times the sojourn time values of the top figure. The diagonal elements are similar in Fig. 1 for all four Markov-chains in both figures as they are fixed by the input (see Eq. 3).

$$A = \begin{bmatrix} 0 & \cdots & \cdots & \cdots & 0 & \pi_2 & 0 & \cdots & \cdots & 0 & \cdots & \pi_n & 0 & \cdots & \cdots & 0 \\ \pi_1 & 0 & \cdots & \cdots & 0 & 0 & 0 & \cdots & \cdots & 0 & \cdots & 0 & \pi_n & \cdots & \cdots & 0 \\ 0 & \pi_1 & 0 & \cdots & 0 & 0 & \pi_2 & 0 & \cdots & 0 & \cdots & 0 & 0 & \pi_n & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & \cdots & \pi_1 & 0 & \cdots & \cdots & 0 & \pi_2 & \cdots & 0 & \cdots & \cdots & \cdots & 0 \\ & & & & & & & & & & & & & & & \\ 1 & 1 & 1 & 1 & 1 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & \cdots & \cdots & \cdots & 0 & 1 & 1 & 1 & 1 & 1 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 1 & 1 & 1 & 1 & 1 \end{bmatrix}_{2n\times n(n-1)} \qquad b = \begin{bmatrix} (1-p_{11})\pi_1 \\ (1-p_{22})\pi_2 \\ \vdots \\ (1-p_{nn})\pi_n \\ \\ (1-p_{11}) \\ (1-p_{22}) \\ \vdots \\ (1-p_{nn}) \end{bmatrix}_{2n\times 1}, x = \begin{bmatrix} p_{12} \\ p_{13} \\ \vdots \\ p_{1n} \\ p_{21} \\ p_{23} \\ \vdots \\ p_{2n} \\ \vdots \\ p_{n1} \\ p_{n2} \\ \vdots \\ p_{nn-1} \end{bmatrix}_{n(n-1)\times 1} \tag{7}$$

$$min\|Ax - b\|_2 \; s.t. x \geq 0. \tag{8}$$

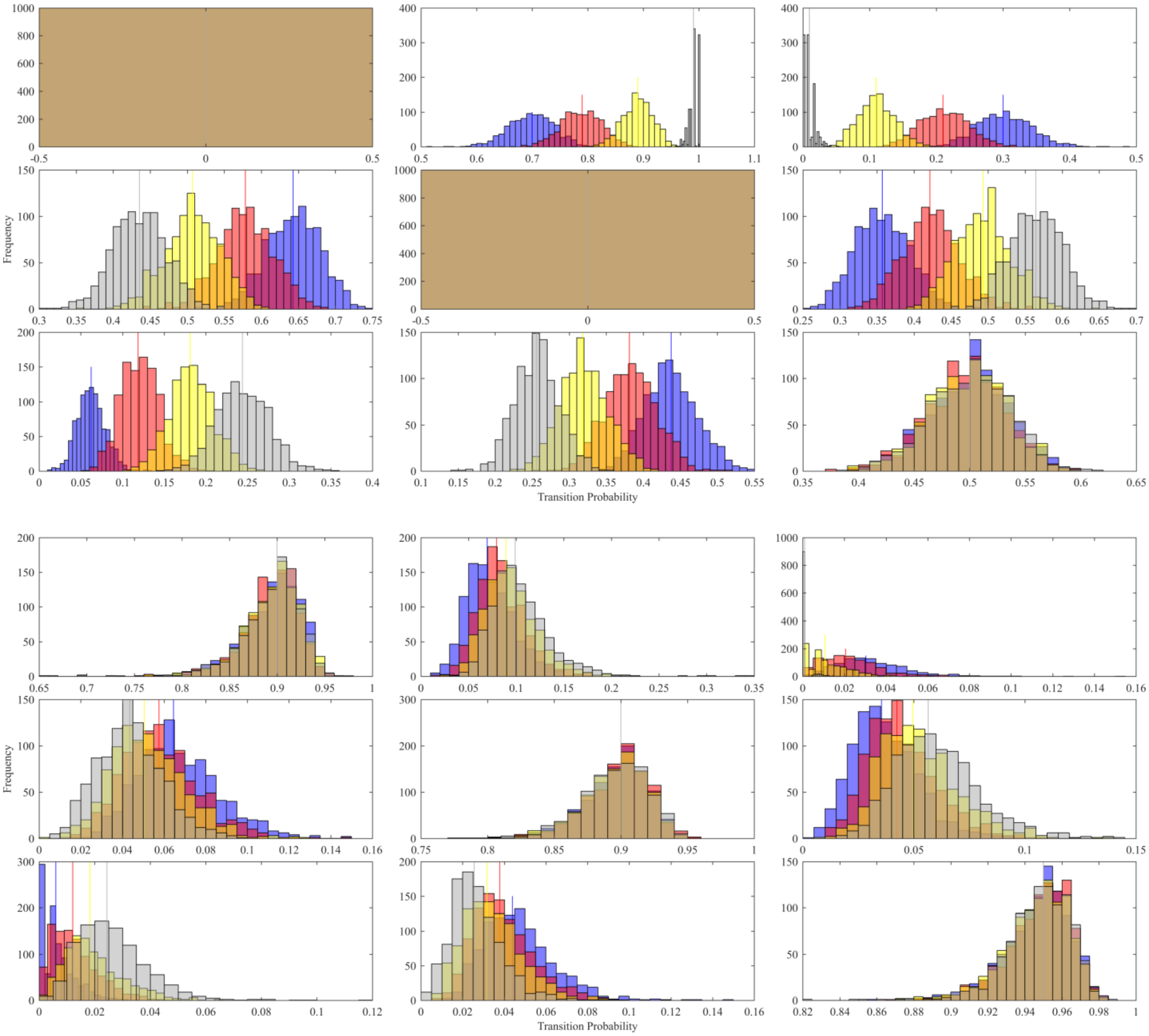

Fig. 1: Histograms of transition probabilities observed from simulation of four (differentiated by colour), 3-state Markov-chains generated using the same stationary probability (π) and sojourn time (τ) inputs.
Top: π = [0.25 0.35 0.40], τ = [1 1 2]; bottom: π = [0.25 0.35 0.40], τ = [10 10 20]. Note, the x-axis scales differ.

For smaller values of sojourn times (Fig. 1, top) the differences in the transition matrices can be clearly observed in their simulated transition probabilities, but the differences are much less when the sojourn times are multiplied by 10 (Fig. 1, bottom). This increase in sojourn times reduces the scale of the distributions of transition probabilities to other states, that is, corresponding to the increase in sojourn times the mean values of the transition probabilities to the other states have reduced by 10. Thus, the differences in the transition probabilities of different Markov-chains are less clear. Thus, for states with a high sojourn time, in absolute terms, the transitions from that state will produce similar results for different transition matrices while those with a low sojourn time will be impacted more strongly by the transition matrix choice.

2.4. Complementing Schedulers

While the Markov-chain scheduler gives a sequence of time-independent activities performed by an occupant, other schedulers are needed to obtain a complete schedule of an occupant's movements through a day. Complementing the Markov-chain scheduler, are three different supporting schedulers: (i) event scheduler, (ii) movement scheduler, and (iii) re-scheduler. Each of these schedulers is accompanied by a location choice model that decides where activities will be performed. Fig. 2 shows the schedulers within the general framework of the movement scheduler for pedestrians (occupants) in office buildings.

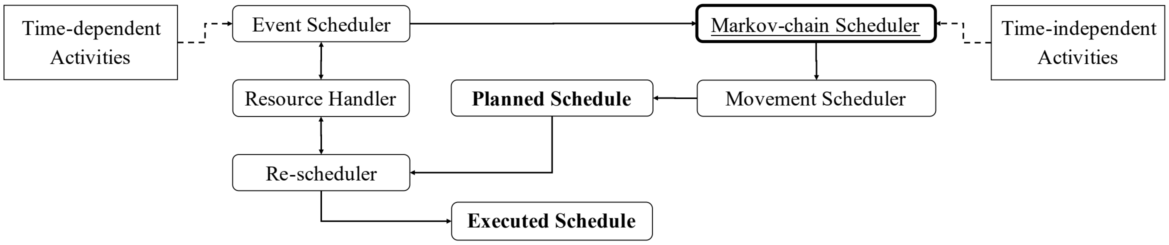

Fig. 2: Framework of the occupant movement scheduler. Single-headed arrows indicate the flow of the movement scheduler; double-headed arrows show interacting parts.

First, the event scheduler is used to schedule time-dependent (first planned and subsequently time-window) activities that anchor the itinerary. Activities such as meetings or arrivals and departures may be generated by sampling from aggregated observations such as meeting room booking data and occupancy sensor data, or expectations of the same. For planned activities, the event scheduler interacts with a resource handler to check the availability of a location chosen by the location choice model before reserving it. Once the schedule is anchored with time-dependent activities, such as a meeting and a lunch break, gaps in the itinerary have to be filled. Since the Markov-chain scheduler derivation holds strictly true only for a large number of time steps, instead of simulating the Markov-chain for the time period of each gap individually, it is simulated for the total period of all the gaps in the itinerary. This long sequence of time-independent activities is then split up to fill in the gaps. The location choice model is used to assign locations where each activity will be performed. Simultaneously, between each activity, the movement scheduler assigns the time required to move from one location to another; thus, developing the planned schedule for a day. Finally, when executing the planned schedule, dynamic updates are required; for example, when a pedestrian discovers that a chosen location is occupied and has to choose a new location, or has to skip an activity because all locations are occupied, or when a delay in starting time of an activity requires shifting the schedule. Location availability is checked by the re-scheduler by interacting with the resource handler before updating schedules based on feedback from the operational level model to result in the final executed schedule.

3. Case Study

A case study of an imaginary consultancy office with a simple layout (Fig. 3) and 18 employees is conducted to face validate the model output. Amongst the 18 employees are 1 manager, 4 senior engineers, 12 junior engineers, and 1 receptionist. All occupants with the same role are assigned the same input profile (stationary probabilities or away times and sojourn times of states). Occupants are assumed to perform the following activities: continuous – being at their desk; recurrent – going to the toilet, getting a drink, taking a break; time-window – arrival, departure, lunch; and planned – meeting. This simple set of activities represents all four categories and is likely to be performed by occupants of most offices.

In order to carry out the first step towards model validation, the following three simulated measures are used: (i) break durations, (ii) time between two episodes (away time) of getting coffee, and (iii) desk occupancy. The first two indicators are compared against the distributions expected from the limiting behaviour of Markov-chain based models while the desk occupancy is judged qualitatively against expected patterns. All measures are calculated as the aggregated output of 10 simulations of the model, that is, 10 days of occupant movements in the imaginary office.

As expected for Markov-chain based models, both distributions show features of exponential distributions. For 10 simulations, the mean break duration is found to be 9.3 minutes while the mean away time for coffee is 84.6 minutes. While the former is close, both values underestimate the true input sojourn times of 10 minutes and 120 minutes respectively. A possible reason for the lower simulated duration may be the splitting of activities from the Markov chain agenda while filling the gaps in the event schedule. Splitting activities would reduce their average duration.

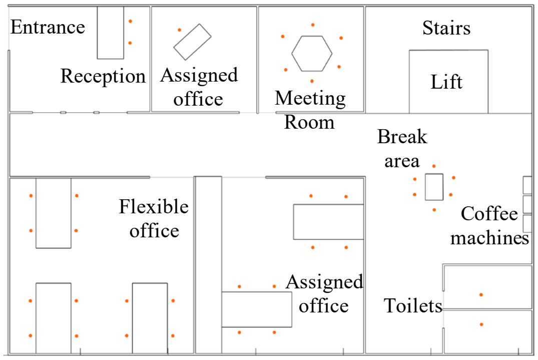

Fig. 3: Layout of the imaginary office used in the case study

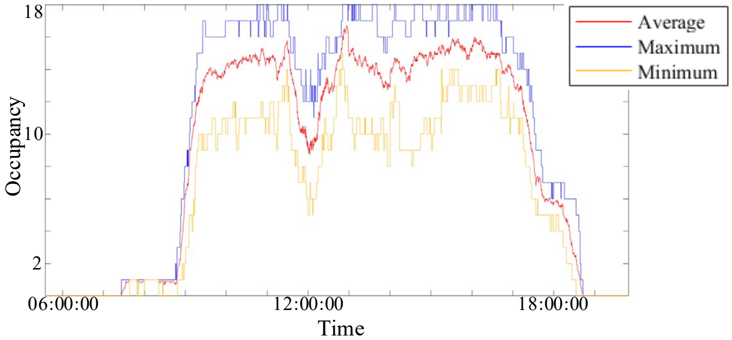

Fig. 4: Minimum (yellow), maximum (blue), and average (red) desk occupancy levels (y-axis) versus time (x-axis) across 10 days

Desk occupancy describes the number of office occupants who are at their desk at a given moment. It is an important parameter for lighting and temperature controls in office buildings because these building services are based on the presence of people in certain areas. Fig. 4 shows the maximum, minimum, and average desk occupancy in a day over 10 simulations. Clear patterns of arrivals at 09:00, lunch break near 12:00, and departures from 16:30 and onwards can be observed. Moreover, the desk occupancy fluctuates throughout the day indicating the natural patterns of people moving about in the office. While these fluctuations seem to be random, as one would expect, the individual activities indeed follow a pattern as indicated by the distributions of the durations of the activities and their away times.

4. Conclusions

While most previous studies have focussed on the operational behaviour level and have not been able to model complex situations for long periods of time, this paper present a tactical behaviour level model that is able to do that. The model can be used to model pedestrian behaviour in office buildings and optimize building services therein. The tactical behaviour level model uses an activity-based framework which produces the high spatial and temporal resolution movement patterns required for testing and optimizing building services in office buildings while the Markov-chain methodology keeps the model data parsimonious and flexible in accepting increasing amounts of information. Furthermore, the paper analyses the transition matrix generation methodology closer than other studies, revealing that using higher temporal resolution – that is, increasing absolute sojourn times – reduces the effect of its under-determined nature on the final result. Finally, the case study of an imaginary office indicates that the model's results could represent movement patterns in an office building.

Despite the advantages of the model proposed here, it has some limitations which could be paths for future studies. To build connections with several studies on workplace interactions in the field of architecture (e.g. [11]), environment-induced activities – activities induced by the current spatial position of the occupant must be modelled. For example, an occupant may find themselves in sight of a colleague and decide to have an unplanned discussion. Furthermore, currently only the first steps towards model validation have been carried out. Therefore, future studies could focus on quantitative validation of the model.

References

[1] S. P. Hoogendoorn, "Normative Pedestrian Flow Behavior: Theory and Applications," Delft University of Technology2001.

[2] A. Schadschneider, H. Klüpfel, T. Kretz, C. Rogsch, and A. Seyfried, "Fundamentals of pedestrian and evacuation dynamics," in *Multi-Agent Systems for Traffic and Transportation Engineering*: IGI Global, 2009, pp. 124-154.

[3] V. Tabak, "User simulation of space utilisation," Eindhoven University Press9068146149, 2008, vol. PhD.

[4] C. Liao, Y. Lin, and P. Barooah, "Agent-based and graphical modelling of building occupancy," *Journal of Building Performance Simulation,* vol. 5, pp. 5-25, 2011.

[5] C. Wang, D. Yan, and Y. Jiang, "A novel approach for building occupancy simulation," *Building Simulation,* vol. 4, pp. 149-167, 2011.

[6] A. Penn and A. Turner, "Space syntax based agent simulation," presented at the 1st International Conference on Pedestrian and Evacuation Dynamics, University of Duisburg, Germany, 2001.

[7] E. Andresen, D. Haensel, M. Chraibi, and A. Seyfried, "Wayfinding and Cognitive Maps for Pedestrian Models," in *Traffic and Granular Flow '15*, Cham, 2016, pp. 249-256: Springer International Publishing.

[8] X. Feng, D. Yan, and T. Hong, "Simulation of occupancy in buildings," *Energy and Buildings,* vol. 87, pp. 348-359, 2015.

[9] Y. Chen, T. Hong, and X. Luo, "An agent-based stochastic Occupancy Simulator," *Building Simulation,* journal article pp. 1-13, 2017.

[10] G. Zimmermann, "Modeling and Simulation of Individual User Behavior for Building Performance Predictions," presented at the Proceedings of the 2007 Summer Computer Simulation Conference, San Diego, California, 2007.

[11] K. Sailer and I. McCulloh, "Social networks and spatial configuration—How office layouts drive social interaction," *Social Networks,* vol. 34, no. 1, pp. 47-58, 2012/01/01/ 2012.