Toward a Multi-Level and Multi-Paradigm Platform for Building Occupant Simulation

Conference Paper · April 2019

8 authors, including:

Davide Schaumann
Rutgers, The State University of New Jersey

Seonghyeon Moon

Muhammad Usman
York University

Rhys Goldstein
Autodesk

Toward a Multi-Level and Multi-Paradigm Platform for Building Occupant Simulation

Davide Schaumann*[1], Seonghyeon Moon*[1], Muhammad Usman[2], Rhys Goldstein[3], Simon Breslav[3], Azam Khan[3], Petros Faloutsos[24], and Mubbasir Kapadia[1]

[1]Rutgers University
Piscataway, USA
{ds1540; sm2062, mk1353}@cs.rutgers.edu

[2]York University
Toronto, Canada
{usman; pfal}@cse.yorku.ca

[3]Autodesk Research
{first.last}@autodesk.com
Toronto, Canada

[4]UHN - Toronto
Rehabilitation Institute
Toronto, Canada

ABSTRACT

In recent years, simulation has been used to investigate building-occupant relations while focusing on pedestrian movement, day-to-day occupancy, and energy use. Most of these efforts employ discrete-time simulation, where building and occupant properties are constantly updated at fixed time steps to reflect building and occupant dynamics. Real-world occupant behavior, however, involves a variety of decision-making patterns that unfold over different time scales and are often triggered by discrete events rather than gradual change. In working toward a platform supporting the full range of human activities in buildings, we embed a discrete-time occupant movement simulator called SteerSuite within a general-purpose discrete-event simulation framework called SyDEVS. With preexisting SteerSuite functions providing low-level steering behavior, and newly implemented SyDEVS nodes providing high-level planning behavior, our prototype represents a multi-level and multi-paradigm approach to occupant simulation for building design applications.

Author Keywords

Multi-Paradigm Simulation; Discrete-Event; Building Occupants; Multi-Level Decision-Making; Discrete-Time.

ACM Classification Keywords

I.6.3 SIMULATION AND MODELING : Applications; I.6.5 SIMULATION AND MODELING : Model Development; I.6.8 SIMULATION AND MODELING : Types of Simulation; J.6 COMPUTER-AIDED ENGINEERING: .

1 INTRODUCTION

Predicting and analyzing the mutual relationship between a building design and the behavior of its occupants is a complex task. In architectural design, architects often use their knowledge and intuition to foresee how a building will impact the movement and activities of its occupants and vice versa. Complete reliance on intuition, however, can bring about a discrepancy between expected occupant behavior and the behavior that actually occurs when the designed environment is built. In some cases, unanticipated behavior can lead to inefficient layouts, occupant dissatisfaction, and wasted energy.

In recent years, simulation methods have been developed to help designers foresee and analyze building-human interactions during the design phase, and thus identify and address design issues before a building is constructed and inhabited. Occupant behavior models developed for building performance simulations (BPS) try to predict how occupants' presence and actions affect, and are affected by, various building systems including heating, ventilation and air-conditioning [28, 14]. Other approaches—some of which stem from computer graphics research—focus on pedestrian movement in emergency and normal operating scenarios [3, 23], as well as day-to-day activities specific to offices [7], universities [22], and hospitals [21].

In most approaches, time advances at discrete time steps. This method is well-suited to approximate the behavior of continuous variables such as the temperature of a building or the movement of an occupant in a space. However, some occupant actions, such as the decision to open a window or to follow a specific route while navigating a built environment, take place at irregular time intervals. These actions are most naturally modeled with a discrete-event approach. Using a discrete-time approach for event-driven actions has a number of disadvantages: (a) some time precision may be lost, since event times may not align with the prescribed time step; (b) some calculations may be redundant, since decisions may end up being evaluated at every time step rather than when necessary; and (c) the integration of multiple models may become

* These authors contributed equally to the work

SimAUD 2019 April 07-09 Atlanta, Georgia

more difficult, since no one time step is optimal for all solvers [10, 20].

To address these issues, we prototype a multi-level and multi-paradigm approach that couples a discrete-event framework based on the Discrete Event System Specification (DEVS) formalism with a discrete-time simulator. The DEVS framework we use is SyDEVS, an open source C++ library featuring base classes from which modelers can derive nodes representing systems and processes in essentially any domain. To demonstrate the type of model composition we envision for occupant-building simulations, we prototype a set of nodes representing building thermodynamics, human comfort, and high-level occupant decisions. The discrete-time simulator is SteerSuite [23], an established crowd simulator. The objective of combining these libraries is to provide a holistic, modular, and extensible model of building occupancy that covers multiple domains and captures behavioral patterns unfolding at different time scales.

Compared with prior work [9], our approach includes a multi-level representation of occupant behavior which accounts for the following: (a) *higher-level* discrete-event decision-making (e.g. defining an agent's movement target) that occurs when specific conditions (events) are triggered (e.g. when the agent's comfort threshold is surpassed); (b) *lower-level* discrete-time representations of phenomena that vary continuously over time, including a building's physical properties (e.g. temperature) and occupant movement, which is influenced by the built environment as well as the dynamic presence and movement of other agents' in the same space.

The demonstrated multi-level and multi-paradigm approach holds promise to enable architects and engineers to integrate independent simulation methods into a shared platform to analyze how a building design will affect its future occupants, how the occupants will affect the building, and ultimately how the overall system will impact the natural environment.

The paper is organized as follows. First, we review existing building occupant modeling and simulation approaches. Then, we introduce our multi-paradigm and multi-level prototype. Next, we demonstrate our approach using a case study. Finally, we draw our conclusions and outline the benefits and limitations of our approach.

2 APPROACHES FOR OCCUPANT SIMULATION

One of the most important challenges that architects, engineers, and building owners face when designing a building is to foresee and analyze the mutual relations between a built environment and the movement and activities of its occupants. This is a complicated task, due to the dynamic, stochastic, and context-dependent nature of human behavior, which both affects and is affected by the built environment as well as the presence and behaviors of other occupants.

To address this challenge, a plethora of simulation methods have been developed in recent years to investigate different aspects of building-occupant interactions. These methods can be classified in a number of ways, including by level of abstraction and by modeling paradigm. We observe three commonly used levels of abstraction: *aggregate*, *planning*, and *steering*. Aggregate models track the utilization of various spaces, but do not represent individual occupants. Planning models track individual occupants, but only capture the high-level decisions that govern which spaces occupants inhabit, which routes they take, and what actions they perform with some degree of deliberation. Steering models also track individuals, but focus on detailed movement and capture low-level decisions such as where to step and how to avoid collisions. Separate from these three levels are two paradigms: discrete-time simulation and discrete-event simulation. The discrete-time paradigm is the more common of the two, and involves fixed time steps at which the state of the represented system is updated. The discrete-event paradigm involves the repeated advancement of time to the next event, generally resulting in variable time steps [9].

Table 1 is a matrix that intersects the three observed levels with both paradigms. This classification strategy creates six categories, and the table lists the most prominent form of occupant simulation in each of them.

	Discrete-Time Paradigm	Discrete-Event Paradigm
Aggregate Level	Building-Centric Hourly Profiles	Building-Centric Survival Models
Planning Level	Discrete-Time Markov Chains	Discrete-Event Multi-Agent Models
Steering Level	Discrete-Time Crowd Simulation	Discrete-Event Movement Models

Table 1. Classification of occupant simulation methods. Highlighted cells indicate the approaches used in this work.

Among the simplest occupancy models are what we refer to as building-centric hourly profiles. With this method, various spaces in a building are each assigned a profile giving the expected number occupants for each hour of the day. The most prominent examples are the profiles provided by ASHRAE [1] and subsequent versions of Standard 90.1. These models are nearly ubiquitous in energy modeling practice, though tools exist to instead employ more sophisticated survival models and Markov Chains [8].

Survival models are loosely based on those that estimate the lifetime of a specimen or entity. Building-centric survival models can be used to simulate the time until the number of occupants in a space changes. This research area began with observations of single-person offices performed by Wang et al. [27]. The more recent work of Parys et al. [20] is informed by a number of preceding survival models in the building performance simulation field.

Various works on discrete-time Markov Chains begin to introduce the concept of tracking individual occupants into energy modeling research. In these models, occupants' transitions from one state to another are based on probabilities, which are examined at every time step. The model of Page et al. [19] only recognized each occupant's presence or absence in a space. Wang et al. [26] use an enhanced version of the method to track occupants from one space to another.

Discrete-event multi-agent models also track individual occupants as they move through a built environment, but each occupant remains in its current state until the next event occurs. There are no fixed time steps at which all occupants are updated. Instead, occupants are treated asynchronously with respect to simulated time. In an example by Goldstein et al. [7], a gamma distribution is used to randomize the time each occupant spends on each task before transitioning to a new activity in a new location. The mathematics is similar to the survival models described above, except that time durations are calculated for each occupant instead of each space. Zimmermann [30] provides another example of occupants modeled as agents with highly asynchronous behavior.

Discrete-time crowd simulations model the flow of pedestrians through a built environment. A variety of techniques are used to predict the dynamics of human behavior in crowd situations. Some of these techniques capture human movement at a very fine level of abstraction; an example is the work of Kapadia et al. [15], which accounts for individual footsteps. Some works employ coarser approximations of the human form, and strive to support large crowds [12]. The majority of implementations employ fixed time steps, which simplifies mechanisms for avoiding collisions. Crowd simulation is used for design applications in industry [17].

Discrete-event movement models have been explored in a few research efforts. Buss and Sánchez [2] provide a complete description of piecewise linear object movement where events correspond with trajectory changes. Another simple example of discrete-event movement arises when agents move at a constant speed on a grid [5], as a diagonal step should take roughly 40% longer than a step to an adjacent grid cell.

Not all research efforts involving occupant simulation fall cleanly into any single one of the above categories. Schaumann et al. [21] investigate narrative-based modeling approaches where workplace procedures involving multiple locations and agents are modeled explicitly. In this work, the choice between discrete time and discrete event is of secondary importance, as the greater challenge is how to specify and recreate the complex collaborative activities that unfold in process-driven facilities like hospitals and factories.

Our interest lies in the pursuit of complex yet scalable occupant models that combine the above mentioned approaches. We focus on the integration of discrete-event multi-agent models for high-level "planning" decisions, with discrete-time crowd simulation models for low-level "steering". This combination spans multiple abstraction levels and both paradigms. There are various techniques for integrating different types of simulation models [4]. A popular one is co-simulation, where multiple simulation engines are run simultaneously and exchange information over time. A relevant example of co-simulation is the occupant behavior modeling tool by Hong et al. [13], which enables co-simulation with building energy modeling software using a functional mock-up interface (FMI). We adopt a more classic formal modeling approach where models are implemented with a common interface, allowing them to be combined hierarchically and coordinated by a single general-purpose simulator [25]. Importantly, this classic approach does not preclude one from making use of preexisting simulation code. In fact, the multi-level and multi-paradigm aspects of our prototype are achieved by integrating two independently developed simulation libraries: the SyDEVS discrete-event framework and the SteerSuite discrete-time crowd simulator. SyDEVS provides the coordinating simulation engine and SteerSuite's capabilities made available by wrapping key parts of the API in a SyDEVS node.

3 A DISCRETE-TIME AND DISCRETE-EVENT PLATFORM FOR BUILDING OCCUPANT SIMULATION

We prototype a multi-level and multi-paradigm platform that couples a discrete-event framework with a discrete-time simulator. High-level occupant decisions (e.g. the next location to visit) are treated using a discrete-event approach, while low-level behaviors (e.g. how to get to the chosen location) are represented using the discrete-time paradigm. Both high-level decisions and low level behaviors impact and are impacted by dynamic environmental conditions, such as the current temperature in the building. For example, occupants' presence contributes to increased building heat. Excessive heat, however, can cause other occupants to move to a different location. A decrease in the number of agents in a room, in turn, will likely lead to a gradual reduction in air temperature.

3.1 Conceptual framework

Figure 1 provides an overview of our conceptual framework. Building data (e.g. building geometry and material properties) and occupant data (e.g. number of occupants, velocities, and initial targets) are used as input for a simulation phase. In this phase, a dynamic building status (e.g. temperature) and an occupant status (e.g. thermal comfort) are updated over time while accounting for their influence on one another. Both statuses inform an occupant behavior calculation system, which is composed of the following components. A high-level discrete-event decision-making system determines the next action that an occupant should perform (e.g. move to a specific target). These high-level actions occur not at every time step, but when a specific event occurs (e.g. the occupant temperature is above a specific threshold). A low-level discrete-time steering algorithm calculates an optimal path to reach a chosen target while avoiding obstacles and accounting for the movement of other agents. Agent movement thus affects the status of the building and the occupants which, in turn, may trigger additional high-level decisions. The simulation results can be visualized at discrete-time steps or when the simulation is complete.

3.2 SyDEVS: A Discrete-Event simulation platform

The Discrete Event System Specification (DEVS) formalism is a set of conventions for representing essentially any discrete event system [29]. The rationale for using DEVS is to support a modular and hierarchical approach to model development while ensuring all time advancement patterns are accommodated. There are a number of simulation frameworks based on DEVS or one of its variants. The framework we use is an open source C++ library called SyDEVS (`https://autodesk.github.io/sydevs/`).

SyDEVS nodes can be of two types: *function* nodes or *simulation* nodes (Figure 2). *Function* nodes are the basic type of

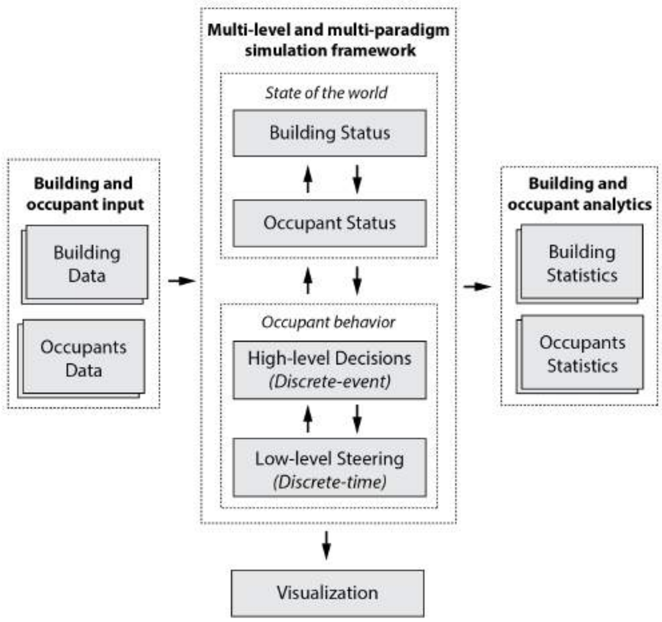


Figure 1. Conceptual framework for a multi-level and multi-paradigm occupant behavior simulation.

dataflow node. They represent a single function that handles one flow event. This function reads a set of input values and calculates a set of output values. *Simulation* nodes represent behavior that unfolds over simulated time. They handle the following types of events: *Initialization Events* are invoked once at the beginning of the simulation; *Unplanned Events* are invoked every time a message is received at unexpected times; *Planned Events* are scheduled by the node; and *Finalization Events* are invoked once at the end of the simulation.

Simulation nodes can be *Atomic*, or can be organized in hierarchical compositions. *Collection* nodes contain any number of instances of an atomic node. *Composite* nodes contain networks (dataflow + DEVS + dataflow) of other nodes, which can themselves be composite nodes, thus forming a hierarchy.

Different types of simulators can be encapsulated within SyDEVS nodes to create modular, hierarchical and extensible data workflows that operate at different time scales. Because the framework employs a multiscale time representation [6], models requiring dramatically different levels of time precision (e.g. seconds, days, femtoseconds) can be linked together and allowed to interact.

3.3 SteerSuite: A Discrete-Time crowd simulator

SteerSuite is an open source C++ framework for crowd simulations (`http://steersuite.eecs.yorku.ca/`). It simulates multi-agent navigation and steering in built environments while responding to the dynamic presence and movement of other agents in space. SteerSuite includes the infrastructure required by typical AI and steering algorithms (i.e. a simulation engine, a spatial database, planning functionality and classes to read and write simulation recordings). It thus facilitates the development of new steering algorithms or the

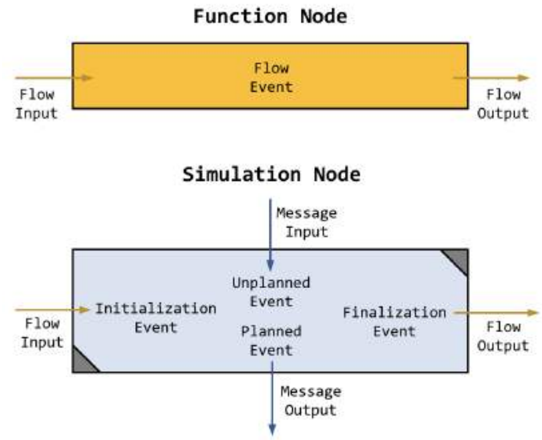


Figure 2. Node types in SyDEVS that can be combined into larger, hierarchical node networks.

use of the following established steering approaches: (a) PPR [23] combines reactions, predictions and planning in one single framework, (b) ORCA [24] uses reciprocal velocity obstacles for goal-directed collision avoidance, and (c) SF [11] uses social forces for resolving collisions between interacting agents in dense crowds. Additionally, SteerSuite visualizes real-time or pre-recorded simulations in 3D environments, and provides built-in modules to analyze the results with respect to a set of customary or user-defined benchmarks.

3.4 Integrated platform

The proposed platform couples the functionality of SyDEVS and SteerSuite to define an integrated framework for multi-level and multi-paradigm occupant-behavior simulation. While SyDEVS supports the modeling of potentially any type of system, in this prototype we have created a specific node composition that demonstrates the multi-level and multi-paradigm nature of the approach. Figure 3 shows an overview of the platform using the SyDEVS notation [16].

The platform consists of a SyDEVS composite node that contains the following data workflow. In an *initialization* phase, a series of function nodes specify building-related and occupant-related parameters including building geometry, external weather conditions, occupants' initial goals, speed, and direction, and a temperature threshold that, if passed, triggers a high-level occupant decision about where to move next.

This data is used as input for a *simulation* phase, where a combination of atomic nodes (connected by means of an event messaging system) represent dynamic interactions between the occupants and the built environment they inhabit. In this prototype, a "weather" node calculates the outdoor temperature and communicates it to a "thermodynamics" node, which calculates the indoor temperature, while accounting for the occupants' latent heat, modeled in the "heat source" node. The indoor temperature is used to calculate occupants' comfort levels in a "comfort" node.

The temperature perceived by each occupant as well as a temperature threshold defined for each occupant are input to a

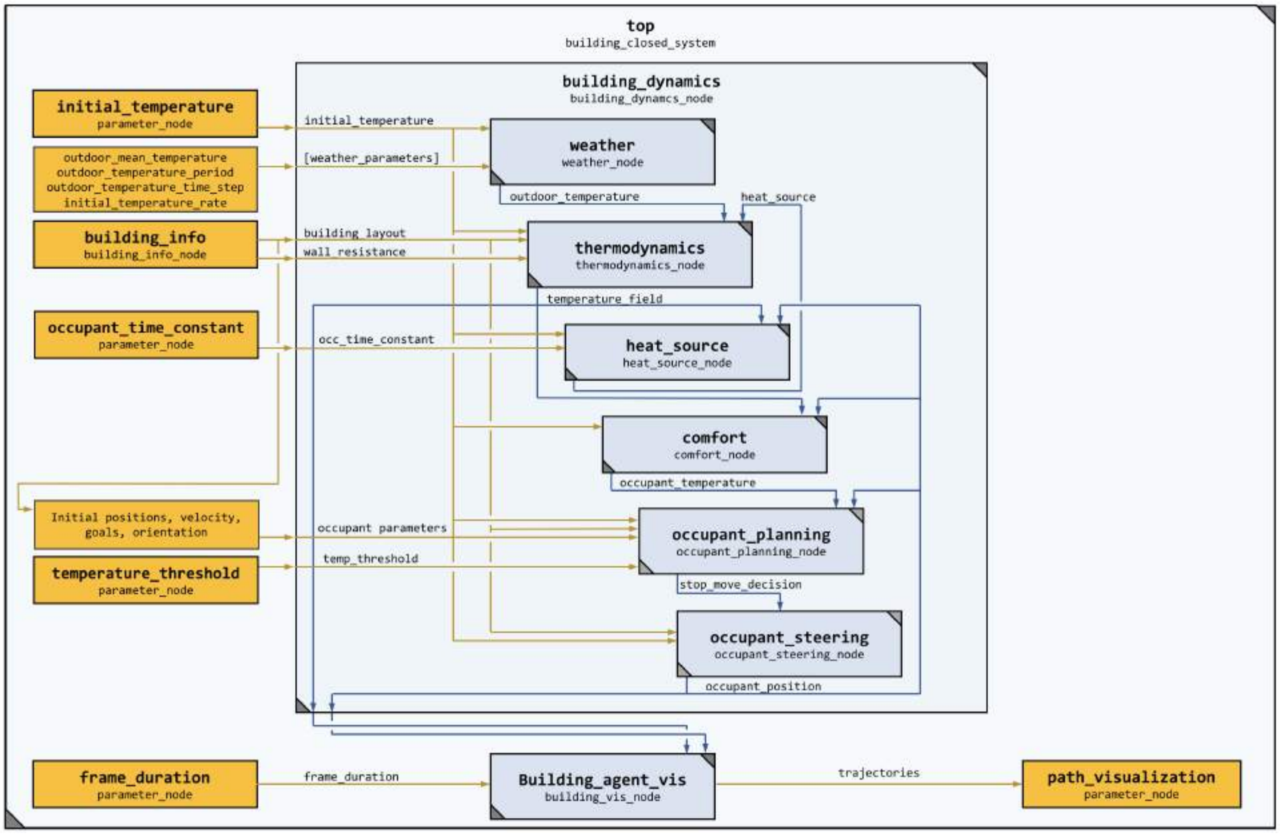


Figure 3. Simulation platform. Multiple SyDEVS nodes are organized into a hierarchical network to represent occupant behavior using a multi-level and multi-paradigm approach.

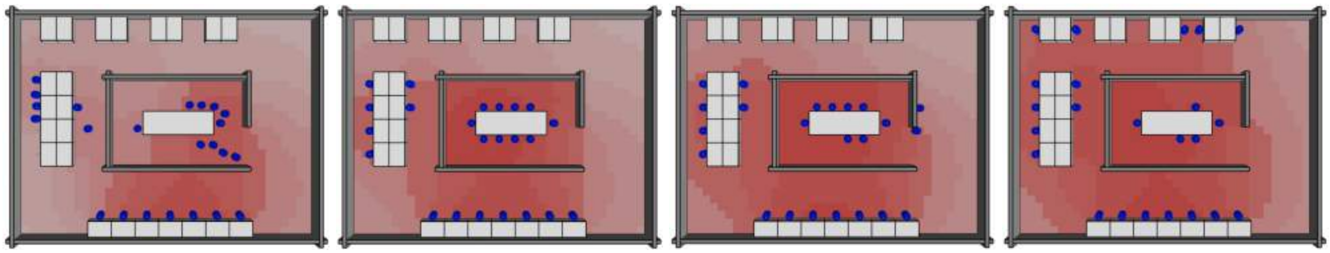
high-level "occupant planning" node, which compares the occupant temperature with its tolerance threshold. If the threshold is passed, the agent is assigned a movement target randomly selected from a user-defined target list. While this node could be modeled as a collection of individual decision making nodes (one for each occupant), in this prototype we implemented a centralized decision-making node that determines which action each occupant should perform. Even though the first approach would enable each occupant to feature an independent decision node that is agnostic to the decision of other agents, the second approach facilitates high-level behavior coordination between multiple occupants, as demonstrated in other multi-agent approaches [21].

The selected occupant's movement target is then communicated to an "occupant steering" node, which encapsulates SteerSuite core functionality. This node calculates agents' movement while accounting for obstacles as well as the presence and movement of other occupants. In this atomic node, a discrete-time approach is used, since agents' path must be re-calculated at specific time steps to account for the dynamic state of the world. The updated occupants' positions are fed back to the previous nodes, which can thus update the indoor temperature and comfort levels, and check whether each occupant's temperature threshold is passed.

A "building agent viz" node collects input from the "thermodynamics" and "occupant steering" nodes to visualize occupant movement and the building temperature over time using SteerSuite functionality. In a *finalization* phase, simulation data is analyzed to represent aggregated occupancy data in the form of movement traces.

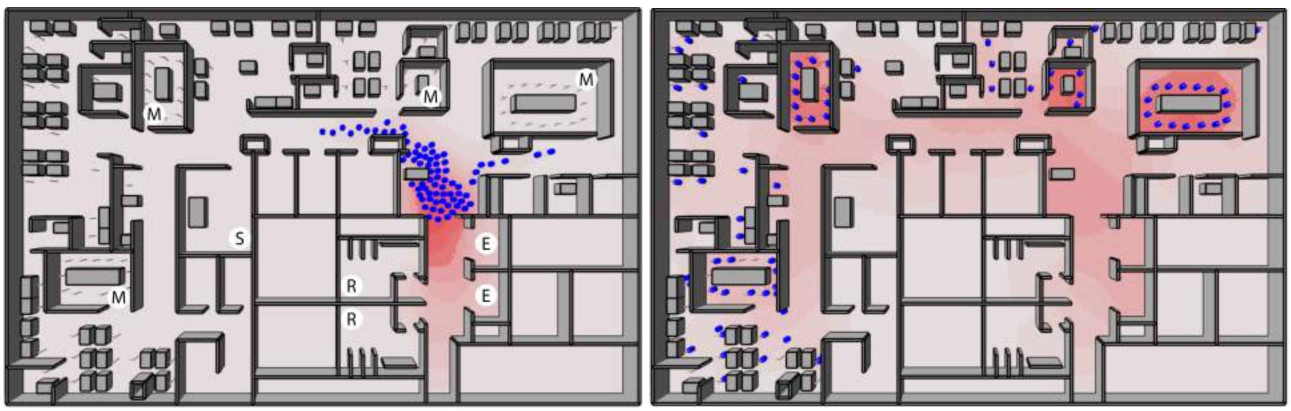
4 CASE STUDY

We apply the proposed platform to simulate the behavior of several occupants in an office space. The building is populated by several occupants, each of which can either work at his/her own desk, or participate in a group meeting. While working, or during a meeting, the simulation monitors the occupant temperature, which is calculated as a combination of the external temperature and the occupants' latent heat. If the occupant's temperature exceeds his/her own specified threshold of tolerance, the occupant will execute a high-level discrete-event decision to change his/her location. In this prototype, we use a low-level discrete-time social force model to calculate agents' steering, although other approaches are supported as well (as mentioned in Section 3.3). In this study, we do not account for occupants' abilities to operate other building systems (e.g. HVAC or windows). However, the proposed framework supports the prototyping of additional nodes that could handle such operations.

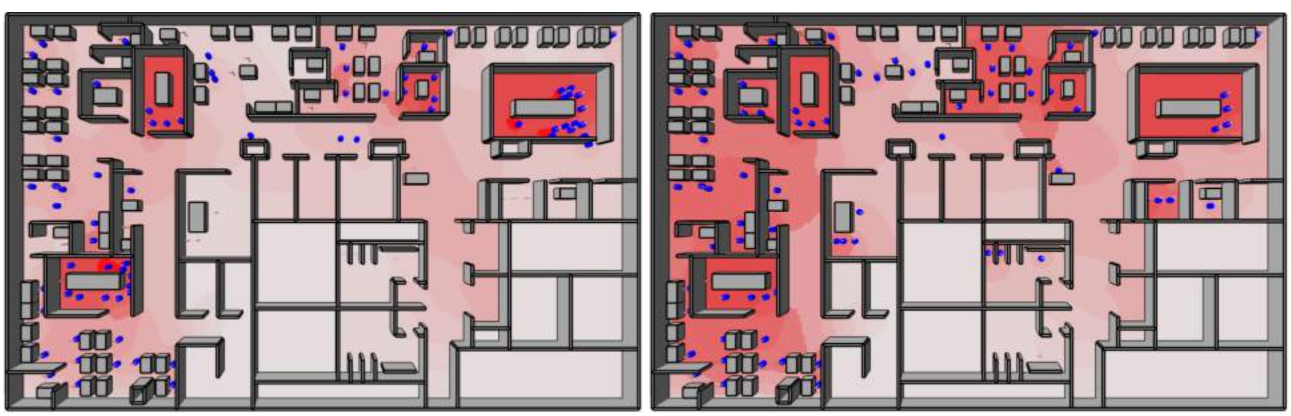


(a) Snapshot at T64 (b) Snapshot at T98 (c) Snapshot at T154 (d) Snapshot at T238

Figure 4. Preliminary study in an abstract office space. A number of occupants are directed either to their desks or to a meeting room (T64). Due to their latent heat, the indoor temperature increases (T98). If the temperature reaches an occupant-defined threshold, the occupant leaves the room (T154). The indoor temperature decreases in the meeting room and it increases in the newly occupied spaces (T238)



(a) Snapshot at T25 (b) Snapshot at T117



(c) Snapshot at T206 (d) Snapshot at T356

Figure 5. Study in an office building, where (E) are the elevators, (M) are the meeting rooms, (R) are the restrooms, and (S) is a social area. 70 occupants are directed either to their desks or to a series of meeting rooms (T25). The indoor temperature increases due to agents' presence and weather conditions (T117). When the indoor temperature reaches an occupant-defined threshold, the occupant is directed either to his/her desk, a social area, or the restrooms (T206). The indoor temperature changes due to occupants' movement to a different location (T356).

Figure 4 shows a preliminary study in an abstract office space. Building occupants are represented in blue, while the current indoor temperature is displayed using different intensities of red (the higher the temperature, the more intense the red). In Figure 4a, the occupants are moving towards their desk or the meeting room. In Figure 4b, a temperature heat map reveals increased indoor temperature in the meeting room. In Figure 4c, some of the occupants leave the room since the current indoor temperature is higher than their threshold of tolerance. In Figure 4d, more occupants leave the meeting room causing

Figure 6. Analysis of occupants' trajectories.

decreased temperature in the room and increased temperature in the newly occupied spaces.

Figure 5 shows a simulation study in a larger office building, populated by 70 occupants that behave in a similar fashion compared to the preliminary study. Some of the occupants go to their desks, while others gather in different meeting rooms. When the occupants' temperature threshold is surpassed, the agents are directed either to a randomly selected targets such as their desks, the restroom, or a social space. Figure 6 displays the aggregated occupants' movement trajectories. A simulation video can be found here: (`https://youtu.be/AgTu6iHD-jc`)

Implementation Details. The building model has been generated in Autodesk Revit, an established BIM tool. The building walls and desks have been exported as XML files using a custom-made exporter. Other building information, such as the weather data as well as wall resistance to heat has been encoded directly into SyDEVS. The initial agents' targets, velocities, orientation and initial positions have also been defined using an XML format and imported into SyDEVS. The visualization of agents' movement and path trajectories has been calculated using SteerSuite.

5 CONCLUSION AND DISCUSSION

The paper introduces initial steps toward a multi-level and multi-paradigm platform for occupant behavior simulation. Our goal is to develop a computational framework that benefits from the advantages of discrete-time and discrete-event representations to capture a wide spectrum of building-occupant interactions that ultimately affect the natural environment. Such an integrated approach would be beneficial since, at a macro level, it could describe the discrete-event decision-making of individual occupants in response to specific environmental changes (e.g. when the indoor temperature reaches a certain threshold) as well as various aspects of the building dynamics. At a micro level, instead, it will be possible to simulate more mundane agents interaction such as movement and obstacle avoidance, which must incorporate constant updates at regular time steps to account for the continuous changes of the state of the world (e.g. the varying position of other agents in a space).

More broadly, our platform enables the simulation of the mutual interactions between building systems and occupant decision-making. In the preliminary studies considered in this paper, we have demonstrated an example of such 2-way interaction. The presence of people in a space affects the indoor temperature; in turn, an excessively high temperature may lead some occupants to leave the room; upon leaving, the indoor temperature may lower down due reduced latent heat produced by the occupant. Other factors that may affect the indoor temperature and the decision-making of the occupant have not been considered in this study (e.g. the operation of other building systems, such as the HVAC).

In our platform, we use SyDEVS as discrete-event framework and SteerSuite as discrete-time steering simulator. SyDEVS features the modeling of node graphs, where each node can encapsulate simulations at different time scales. A specific node composition has been developed for this project. The SyDEVS framework, however, can be fully extended to incorporate additional nodes that extend our current thermal simulation and decision-making systems. Future work will involve integrating the platform with additional nodes, and combining it with an energy simulator, such as Energy Plus. Additionally, we aim to extend occupants decision-making abilities to account for schedules and a more advanced way-finding model that operates at two different levels of abstraction. A high-level decision-making will determine which route, while a low-level decision will calculate more fine-grained movement while avoiding obstacles.

Emerging approaches to design, such as the MaRS Discovery District generative design project [18], test new ways to incorporate human experience metrics into built environments' layouts. In the building science community, researchers are actively exploring how multi-agent occupant simulations could potentially impact energy use predictions [13]. We argue that the proposed approach sets a stepping stone towards a platform that can be used across disciplines to study building-human interactions. Such a platform, could potentially be integrated in architectural design workflows to design settings that maximize occupant needs while minimizing the collective impact on the natural environment.

ACKNOWLEDGEMENTS

This research has been partially funded by grants from the NSERC Discovery and Create programs, NSF IIS-1703883, NSF SAS-1723869, DARPA SocialSim-W911NF-17-C-0098, and the Murray Fellowship. We thank Taewon Kim and Nikhil Jiju for their contribution to this project.

REFERENCES

1. ASHRAE. *Standard 90.1, Appendix G*, 2004.
2. Buss, A. H., and Snchez, P. J. Simple movement and detection in discrete event simulation. In *Proceedings of the 37th Conference on Winter Simulation*, Winter Simulation Conference (2005), 992–1000.
3. Chu, M. L., and Law, K. Computational framework incorporating human behaviors for egress simulations. *Journal of Computing in Civil Engineering 27*, 6 (2013), 699–707.

4. Goldstein, R., Breslav, S., and Khan, A. Using general modeling conventions for the shared development of building performance simulation software. In *Proceedings of the International Building Simulation Conference* (2013).

5. Goldstein, R., Breslav, S., and Khan, A. Towards voxel-based algorithms for building performance simulation. In *Proceedings of the IBPSA-Canada eSim Conference* (2014).

6. Goldstein, R., Khan, A., Dalle, O., and Wainer, G. Multiscale representation of simulated time. *Simulation* (2017), 0037549717726868.

7. Goldstein, R., Tessier, A., and Khan, A. Space layout in occupant behavior simulation. In *Conference Proceedings: IBPSA-AIRAH Building Simulation Conference* (2011), 1073–1080.

8. Gunay, B., O'Brien, W., and Beausoleil-Morrison, I. A toolkit for developing data-driven occupant behaviour and presence models. In *Proceedings of the IBPSA-Canada eSim Conference* (2016).

9. Gunay, H. B., O'Brien, W., Beausoleil-Morrison, I., Goldstein, R., Breslav, S., and Khan, A. Coupling stochastic occupant models to building performance simulation using the discrete event system specification formalism. *Journal of Building Performance Simulation 7*, 6 (Nov. 2014), 457–478.

10. Haldi, F., and Robinson, D. Interactions with window openings by office occupants. *Building and Environment 44*, 12 (2009), 2378–2395.

11. Helbing, D., and Molnar, P. Social force model for pedestrian dynamics. *Physical review E 51*, 5 (1995), 4282.

12. Hesham, O., and Wainer, G. A. Context-sensitive personal space for dense crowd simulation. In *Symposium on Simulation for Architecture and Urban Design* (2016).

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

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

15. Kapadia, M., Pelechano, N., Allbeck, J., and Badler, N. Virtual crowds: Steps toward behavioral realism. *Synthesis Lectures on Visual Computing: Computer Graphics, Animation, Computational Photography, and Imaging 7*, 4 (2015), 1–270.

16. Maleki, M., Woodbury, R., Goldstein, R., Breslav, S., and Khan, A. Designing devs visual interfaces for end-user programmers. *Simulation 91*, 8 (2015), 715–734.

17. Morrow, E., Mackenzie, I., Nema, G., and Park, D. Evaluating three dimensional vision fields in pedestrian micro-simulations. *Transportation Research Procedia 2* (2014), 436 – 441. The Conference on Pedestrian and Evacuation Dynamics 2014 (PED 2014), 22-24 October 2014, Delft, The Netherlands.

18. Nagy, D., Lau, D., Locke, J., Stoddart, J., Villaggi, L., Wang, R., Zhao, D., and Benjamin, D. Project discover: An application of generative design for architectural space planning. In *Symposium on Simulation for Architecture and Urban Design* (2017).

19. Page, J., Robinson, D., Morel, N., and Scartezzini, J. L. A generalised stochastic model for the simulation of occupant presence. *Energy and Buildings 40*, 2 (2008), 83–98.

20. Parys, W., Saelens, D., and Hens, H. Coupling of dynamic building simulation with stochastic modelling of occupant behaviour in offices–a review-based integrated methodology. *Journal of Building Performance Simulation 4*, 4 (2011), 339–358.

21. Schaumann, D., Breslav, S., Goldstein, R., Khan, A., and Kalay, Y. E. Simulating use scenarios in hospitals using multi-agent narratives. *Journal of Building Performance Simulation 10*, 5-6 (2017), 636–652.

22. Shen, W., Zhang, X., Shen, G. Q., and Fernando, T. The user pre-occupancy evaluation method in designer–client communication in early design stage: A case study. *Automation in Construction 32* (2013), 112–124.

23. Singh, S., Kapadia, M., Hewlett, B., Reinman, G., and Faloutsos, P. A modular framework for adaptive agent-based steering. In *Symposium on Interactive 3D Graphics and Games*, ACM (2011), 141–150.

24. Van Den Berg, J., Guy, S. J., Lin, M., and Manocha, D. Reciprocal n-body collision avoidance. In *Robotics research*. Springer, 2011, 3–19.

25. Vangheluwe, H., De Lara, J., and Mosterman, P. J. An introduction to multi-paradigm modelling and simulation. In *Proceedings of the AIS'2002 conference (AI, Simulation and Planning in High Autonomy Systems), Lisboa, Portugal* (2002), 9–20.

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

27. Wang, D., Federspiel, C. C., and Rubinstein, F. Modeling occupancy in single person offices. *Energy and Buildings 37*, 2 (2005), 121–126.

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

29. Zeigler, B. P., Kim, T. G., and Praehofer, H. *Theory of Modeling and Simulation*. Academic Press, Jan. 2000.

30. Zimmermann, G. Agent-based modeling and simulation of individual building occupants. In *Proceedings of SimBuild 2010 conference: IBPSA-USA* (2010), 269–276.