

# Simulating the Sensing of Building Occupancy

Simon Breslav<sup>1</sup>, Rhys Goldstein<sup>1</sup>, Ben Doherty<sup>2</sup>, Dan Rumery<sup>2</sup>, Azam Khan<sup>1</sup>

<sup>1</sup>Autodesk Research

210 King St. East, Toronto, Ontario, Canada, M5A 1J7  
{firstname.lastname}@autodesk.com

<sup>2</sup>BVN Architecture

255 Pitt St, Sydney, NSW 2000, Australia  
{Firstname\_Lastname}@bvn.com.au

**Keywords:** occupancy simulation, sensor simulation, DEVS

## Abstract

Accurate building occupancy information can be beneficial in minimizing energy use by improving the intelligence of a Building Automation System (BAS) and helping designers predict the effect of different design options on occupant behavior. However, current occupancy measurements are quite inaccurate due to limitations in sensing technology and the resulting discrepancies between sensor data and what actually happens. In this paper we explore the use of simulation to model occupant behavior in combination with motion sensors to be able to study the relationship between known and measured occupant behavior. An extensible occupancy model, influenced by computational cognitive science and implemented using established modeling conventions is presented along with a simple experiment comparing the effects of different sensor density levels.

## 1 INTRODUCTION

It is very difficult to predict the rich tapestry of activity performed by building occupants through their day-to-day actions. As a result of this difficulty, lighting, heating, air-conditioning, and many other devices are left activated to consume power even when people are absent. Benefits of power-saving technology cannot be quantitatively assessed due to these large unknowns related to occupant behavior. Furthermore, it is difficult to compare different design options for their probable impact on occupant satisfaction, comfort, presence in various rooms, and use of water and electricity. These are tremendously important issues, since building operations account for the largest portion of global energy consumption and greenhouse gas emissions (US EIA, 2008).

Instrumenting our built environment with sensors and collecting detailed occupant data can address these issues, but this is complicated by the unavoidable discrepancy between sensor data and what actually happens. For example, a very common sensor technology for detecting occupancy is Passive Infrared (PIR) motion sensing. For this kind of sensor the discrepancy between sensed and real data occurs when certain areas are not covered by the sensors, since they have a limited range and obstacles such as furniture can interfere with

proper detection. Also, even if the person is within the range of the sensor, if the person remains too still the sensor may not detect the person's presence. Treating occupant sensor data as a truthful representation of actual occupant presence could be detrimental to an intelligent Building Automation Systems (BAS) and to a designers' understanding of human behavior in buildings.

In this paper we present work towards the use of simulation to study the relationship between actual occupant behavior and occupant behavior measured by sensor networks in buildings. The simulation encompasses a building with different rooms that have sensors within them. Occupants of the building possess high level goals such as drinking water and checking email, which they satisfy by walking around a building and performing actions. As occupants walk around the building they trigger motion sensors. In our experiment we focus on evaluating different configurations of PIR-based motion sensors by comparing actual occupancy within each room to occupancy sensed by sensors.

A major goal of our work is to establish a framework to be used in future research. To set a solid foundation in that direction, the organization of the occupancy model has influences from computational cognitive science research. And most importantly, we have adopted a simulation convention called Discrete Event System Specification (DEVS) (Zeigler et al., 2000) to set a solid theoretical foundation and allow for better extensibility and future improvement of our simulation models. Throughout the paper we will point out some of the positive effects that adherence to the DEVS conventions has provided.

## 2 BACKGROUND

There has been a large volume of work on occupant simulation in different fields of research, including computer graphics, architecture, and energy modeling, to name a few. However, in practice, these simulations are rarely utilized for energy-efficient building design. Instead, fixed occupancy profiles are used for most building energy models. These fixed profiles are often based on large-scale metering data from utility companies or building energy codes and standards (Chiou, 2009). Such fixed profiles are not sufficient to realistically capture the relationship between a building's energy consumption and individual occupants' behavior, since

no distinction is made between individual occupants, no interaction occurs between groups of occupants, and there is little day-to-day variability. To fully understand the consequences of individual choices that occupants make, more detailed occupant models are needed.

There are many applications of occupant simulation in the field of computer graphics such as film, video games, training, and emergency evacuation. Virtual Crowds (Pelechano et al., 2008) surveys work in this area, establishing a baseline of techniques and requirements for simulating large-scale virtual populations of seemingly sentient beings that possess both crowd and individual goals. We hope to build from this computer animation perspective to achieve a greater understanding of human behavior as it relates to comfort and energy consumption in buildings.

In architecture, there has also been some push to use computer-aided design tools to look at how space is used by occupants, instead of just looking at architecture only in terms of geometrical form. As stressed in the Space Re-Actor thesis (Narahara, 2007), computational design tools need to go beyond visualizations of light, materials and geometry, and help people study and understand designed spaces as they are to be inhabited. In his thesis Nagukura presents a system where agents display different behaviors in reaction to architectural elements, such as glass, perforations, and furniture.

When it comes to behavior models, the most interesting and advanced models are being developed in computational cognitive science in an effort to merge neuroscience, artificial intelligence, and cognitive psychology to produce models of human cognition (Vernon et al., 2007). While our occupant model is not a direct implementation of any of the existing cognitive architectures, we are inspired by the current work in the field, and hope to integrate more aspects of these systems in the future.

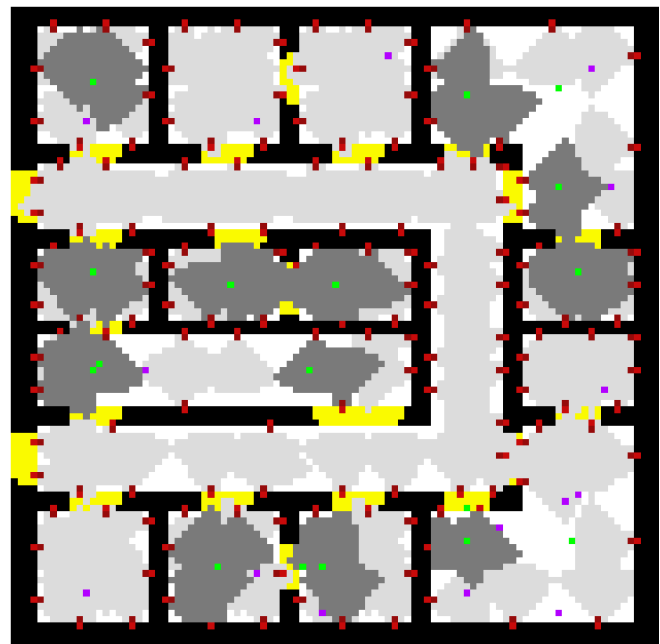
Using more advanced occupant models instead of fixed profiles in the context of building energy modeling can have a significant impact on energy consumption predictions. For example, (Bourgeois et al., 2005) showed that energy use predictions changed by 62% in an office building when the simulated occupant had the ability to manipulate heating, cooling, and lighting controls. Another example by Hoes et al. (2009), showed that the use of a detailed occupant model can have a significant effect on both heating and cooling energy demand predictions.

To study the relationship of actual occupant behavior, and behavior collected by sensors, we need to integrate sensor networks into our agent framework. Modeling sensors and occupants in combination to find relationships between these two aspects has not received much attention in the context of building design and operation. Sensor networks are an important aspect of surveillance systems, traffic control, and health-care monitoring, and there has been a number of papers on au-

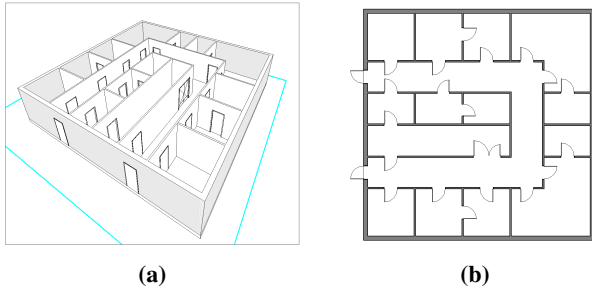
tomatic sensor placement (Becker et al., 2009; Hengel et al., 2009; Garaas, 2011). This previous work focuses on finding ways to fully cover the geometric space with camera sensing. In our work, on the other hand, we are interested in evaluating, and understanding the relationship between sensor placement and accuracy of occupant measurement. To achieve that kind of understanding, we need an advanced occupant behavior model where people are able to perform actions such as walking around the building along paths for sensors to monitor, and in the next section we will discuss the details of creating such a model.

### 3 MODEL

The goal of our experiment is to study the relationship between actual occupant behavior within each room and occupant behavior measured by motion sensors within a building. To achieve that, our simulation involves occupants walking around a building performing actions to satisfy their goals, while motion sensors detect occupant presence. To understand the mechanism of the simulation, in Section 3.1 we will introduce the overall process in terms of its basic elements and output behavior. Then, in Section 3.2, we present the details of our occupant model. We conclude with ways we have used DEVS conventions for the model in Section 3.3.



**Figure 1:** Grid-based floor plan. Green dots are occupants, purple dots are waypoints, red dashes are sensors with the direction of the line indicating the orientation. Light gray areas on the floor indicate the area monitored by the sensor, and dark gray areas indicate that there is an occupant within that sensed area. Doors are in yellow.



**Figure 2:** Input building geometry, where (a) refers to the original 3D floor plan, and (b) to the extracted 2D floor plan.

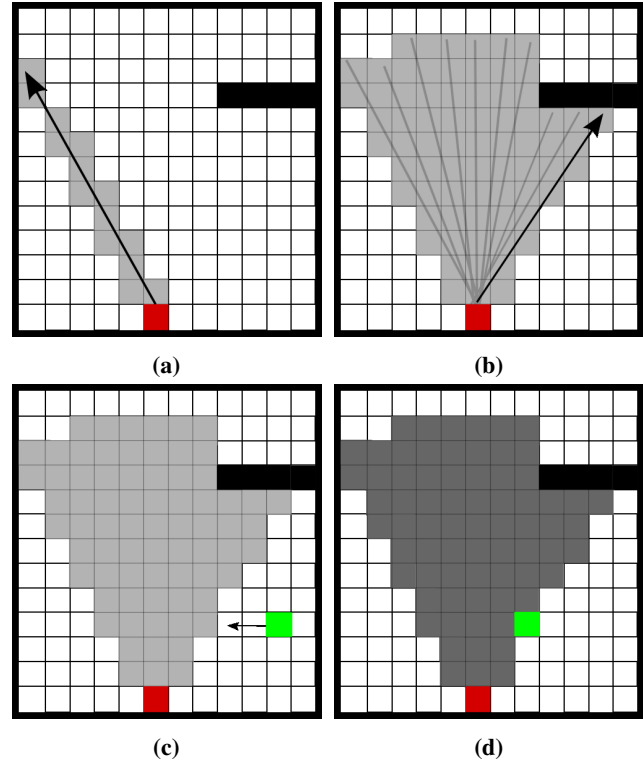
### 3.1 Building Model

The main elements of the building model are the space layout, the occupants, and the sensors. The space layout remains fixed throughout the simulation while occupants move around it. As an occupant walks into a detection area of a sensor, that sensor gets activated, and when the person leaves the detection area, the sensor is de-activated.

Our space layout is represented as a discrete 2D grid of cells (see Figure 1). To create such a grid, a 2D floor plan (see Figure 2b) is extracted from a 3D building (see Figure 2a) which is discretized into cells. Different cells represent different elements, such as open areas, obstacles, doors, sensors, occupants, and others. The cells are colored based on their element type (Figure 1). People move from cell to cell based on the rules described in their behavior model. We have adopted a grid-based space layout representation for a number of reasons, such as to increase computational speed and to simplify the programming of the model. Alternative ways to represent the space layout, such as full 3D geometry, may be explored in future work. To generate the grid-based representation from an Autodesk Revit model of the building, we used the Spatial Analysis and Query Tool (SQ&AT). SQ&AT is a Revit plugin that affords the interactive analyses of circulation patterns of buildings (Doherty et al., 2012).

Besides the space layout data, *Waypoints* are an important input for the simulation. A *Waypoint* is a reference point in the building used for purposes of navigation. In our simulation we restrict occupants to move between these *Waypoints*, reflecting the observation that in most rooms, occupants tend to spend time at certain positions. These locations are manually authored during the design of the building, and would normally coincide with furniture or electronic devices used by the occupant. In Figure 1 they are illustrated with purple dots.

The occupant behavior is driven by high level goals such as reading an email, drinking water, or eating lunch. To achieve a high level goal, an occupant breaks the goal down to actions that she can perform. In our simulation all goals break down



**Figure 3:** Sensor Detection Process. (a) A ray is shot from the sensor (red cell), and as the ray travels through the grid space, grid cells are marked as belonging to the sensing area (gray cells). The ray stops if it hits a wall or if it reaches maximum range. (b) The process repeats for a number of rays to form a cone shaped area. (c) When an occupant (green cell) is outside the sensing area, the sensor is inactive. (d) When an occupant is inside, the sensor becomes active and the sensed area turns dark gray.

into at most two actions; first walking to the location where the goal can be satisfied, and second, actually performing the action necessary, e.g. drinking water. The locations at which the goal can be satisfied is picked by the selection of a random *Waypoint*. In future experiments we intend to make both of these procedures much more involved and context dependent. In Section 3.2, we will describe the *Occupancy* model in more detail.

PIR-based motion sensors detect occupant presence by sensing the difference between infrared light emitted by moving people and the background. We have modeled each PIR sensor as an emitter of rays at a given location and direction. The term passive in this instance refers to the fact that PIR devices do not generate or radiate any energy for detection purposes. However, for computational and implementation efficiency, instead of modeling walls and occupants as sources of radiation, we trace rays from the sensors outwards. In Figure 1 sensors are illustrated using red line with the direction of

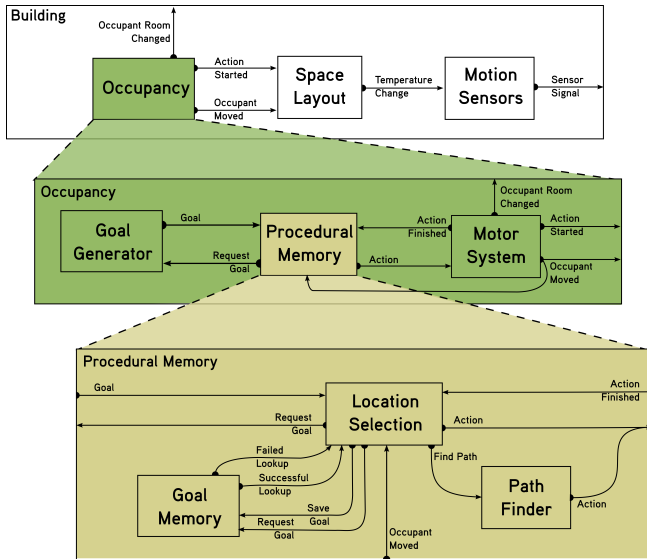


Figure 4: Building Simulation Model.

the line indicating the orientation of the sensor, with all of the sensors being mounted on the walls. To find the sensed area for a given sensor, rays are sent out using a fast grid-based ray tracing algorithm (Amanatides and Woo, 1987). See Figures 3a and 3b for the illustration of this process. Ray casting is necessary to make sure that sensors are blocked by walls and other obstacles. In practice, we perform the ray casting only once, during the initialization of the simulation, and cache the results in a reference grid for the whole building. This caching allows for fast lookup if an occupant passes within the detection range of the sensor. An important parameter for the *Motion Sensor* model is the sensor's range which includes the distance (8 meters) and horizontal angle range ( $90^\circ$ ) at which the sensor can perform detection. The process of sensing the occupant is illustrated in Figures 3c and 3d.

### 3.2 Occupant Model

Our overall goal in designing the *Occupancy* model was two fold; first, keeping it as simple as possible to achieve the desired simulation, and second, to set a clear foundation for future extension to more complicated behavior. Similar to other agent-based work (Yu and Terzopoulos, 2007), our occupant model is composed of different conceptual components. To talk about these components in a way that clearly communicates their purpose, we use fundamental cognitive psychological themes consistent with work in computational cognitive science (Vernon et al., 2007).

The submodels of our *Occupancy* model fall under the following themes:

**Motivation** A *Goal Generator* model creates a goal for the occupant to perform. An example of a goal is to satisfy

hunger or thirst. A goal has an associated duration for the goal to be satisfied. Currently, this duration is a random time drawn from an exponential distribution with a mean of 10 minutes.

**Motor System** is responsible for controlling movement and is represented by a submodel called *Motor System*. It knows how to walk and execute various actions such as drink and eat.

**Procedural Memory** is memory for how to perform actions. In our model the responsibility of *Procedural Memory* is to take a high level goal and translate it to low level actions that the motor system can understand and perform.

**Short-Term Memory** is memory that can hold information for a short period of time. *Goal Memory* is responsible for keeping track of future goals that the occupant needs to satisfy. It is implemented using a simple queue. For example, if an occupant's goal is to drink water in room 8, but she is located in room 7, the *Goal Memory* would save the goal of drinking water in room 8, while the occupant walks there.

**Spatial Memory** is the part of memory responsible for information about one's spatial location and orientation. In our simulation two models fall into this theme, *Location Selection* and *Path Finder*. *Location Selection* is a model responsible for selecting the location of a given goal. In our simulation we simply pick random *Waypoints* from the building. One can imagine extending this to pick only locations depending on the goal at hand, for example, washing hands should only be done at a sink. However, for our experiment, such extensions were not necessary. *Path Finder* is a model which returns the shortest path between two points, respecting all the walls and obstacles in the building. This model uses the A\* Algorithm (Hart et al., 1968) for finding the shortest path, which is very popular in the computer game industry due to its flexibility in a wide range of contexts by offering a tradeoff between speed and accuracy. It also allows different areas of the building to have different costs associated with them, to allow encouragement or discouragement of occupants from going one way or another. For example, areas in a hallway right against a wall might have a higher cost value than areas in the middle of the hallway, or an elevator might have a different cost than stairs. The biggest downside of the current implementation is that all the paths are pre-computed at the initialization of the simulation. In the future we hope to expand the simulation to use more adaptive pathfinding.

As described, our occupant behavior is missing crucial pieces of the human cognitive cycle such as perception, attention, emotions, and decision making capabilities to name

a few. However, even with these limitations it does satisfy our goals for the experiment, where an occupant is capable of generating a goal and is able to walk around the building to satisfy that goal. This set of models is also sufficient to set up the framework for future research where each psychological theme can be further expanded.

### 3.3 DEVS Hierarchy

DEVS is a set of conventions for specifying simulation models. The conventions promote modular models that allow for better extensibility. There are two types of models that are defined by DEVS, *atomic* and *coupled*. *Coupled* models are created by linking or coupling other models, while the *atomic* models are indivisible. A *coupled* model can be composed of many *atomic* and other *coupled* models, without needing to know if the models within are *atomic* or *coupled*. One of the important benefits of support for *coupled* models is that it allows one to break the simulation model into a hierarchical set of models. This gives rich flexibility to abstract complicated behavior behind intuitive sets of modules.

Please refer to Figure 4 for the illustration of the *Building* simulation model hierarchy, showing all the models that make up the *Building* and the *Occupancy* model. The links in the diagram represent the pathways of information flow. For example, a Goal is sent from *Goal Generator* to *Procedural Memory*, and within the *Procedural Memory* model it travels to *Location Selection*. *Location Selection* in turn may translate the goal into an Action, and send it out to its parent model, *Procedural Memory*, who will pass it to *Motor System*. However, if an occupant is not at the right location to perform the action, *Location Selection* will send the goal to *Goal Memory* instead, for temporary storage, and will request *Path Finder* to find a path to the desired location. It will send an Action to walk to that location and *Procedural Memory* will delegate the walk Action to *Motor System*. *Motor System* will update the occupant's location and send out all the messages regarding the changes internally to *Procedural Memory* and to the parent *Occupant* model who will delegate the message onward to *Space Layout*. *Space Layout* will then send a message to the *Motion Sensors* model.

In designing our *Occupancy* model, we have created a hierarchy based on cognitive models to stress the idea that the model represents a general behavior of occupancy instead of individual occupants. Thus, an important property of the model is that it represents all the occupants in the building. This approach implies that input and output messages must contain occupant IDs and that some internal variables have to contain structures indexed by this occupant ID (e.g. a dictionary or an array) instead of a single variable. A possible alternative to support multiple occupants is to take advantage of Dynamic Structure DEVS (Barros, 1995), which gives one the ability to define a network of models that can undergo

structural changes during a simulation so that each instance would be a new model.

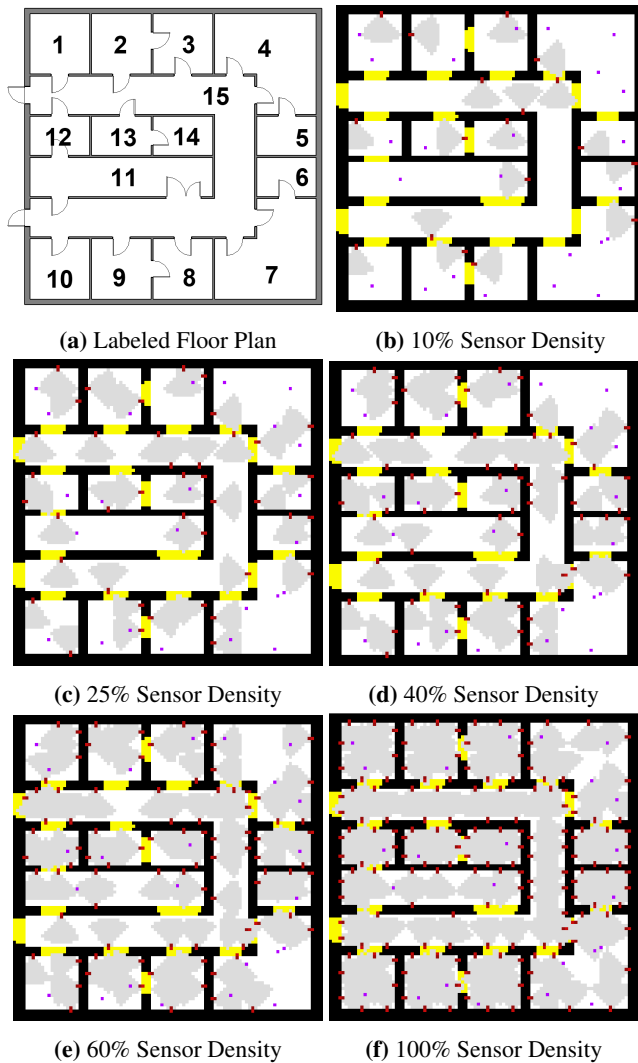
Using DEVS in designing our models allowed us to isolate the responsibility of different aspects of the simulation, where each model does not have to know anything about any other models. This is very convenient, since often the models have to be adjusted or changed. If different models have high dependency on each other, this would be hard to achieve. When models send out messages, they know nothing about the recipients of those messages, and thus stay independent from the surrounding models. This property of reconfiguration is crucial in the context of occupant behavior, since as our understanding of cognitive processes evolve and new computational models for different aspects of the mind are developed, we will be able to adjust accordingly.

## 4 EXPERIMENT

To better understand the discrepancy between occupancy data collected by sensors and true occupancy, the goal of our experiment is to investigate how sensor configuration influences accuracy of occupancy detection. Specifically, our experimental variable is per-room sensor density. Sensor density is the degree at which a given room is sensed, high density implies lots of motion sensors in a room, while low density having few sensors. Refer to Figure 5 for examples of how the density maps to the degree of sensing of the space layout.

To create the full 100% sensor density placement for a given room, each wall segment was instrumented with two sensors. The placement along the wall is chosen by parameterization the wall on a  $[0, 1]$  interval, and placing the sensors at 0.2 and 0.6 locations. This can be seen in 5f, where sensors are represented as red dashes. While this simple sensor placement schema will not scale well for more complicated building plans, it is sufficient for our basic experiment. To produce the sensor density of 10%, for each room, 10% of the sensors are randomly used from all the sensors that were initially placed. Then, to produce 25% density, more sensors are added on top of the ones that were previously added when creating the 10% density configuration. This procedure ensures that sensors that were present in the 10% density trial are also present in the 25% density trial.

For each density (10%, 25%, 40%, 60%, 100%) a trial was run for 8 simulated hours. As the simulation is running, occupants walk around the space layout to get from one *Waypoint* to another, activating and deactivating sensors in the process. The number of occupants in the building is a parameter initialized to generate 30 people. When an occupant gets to their *Waypoint* of interest, they perform an action (e.g. drink water). The time it takes to perform the action is randomly drawn from an exponential distribution with a mean of 10 minutes. This mean time is also an input parameter for the simulation

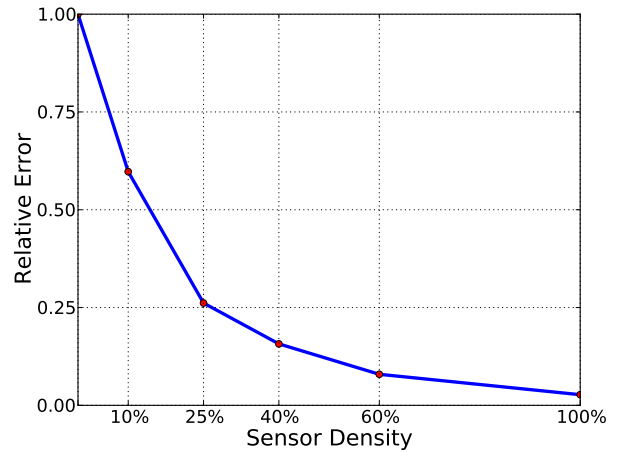


**Figure 5:** Floor Plans with different sensor densities. Red dashes are sensors with the direction of the line indicating the orientation. Light gray area on the floor indicates the sensed area.

set during the initialization phase.

To calculate both the sensed and actual occupancy in each room, we look at times when occupants walk in and out of sensor coverage and enter and exit rooms. According to the sensor network, a room is considered occupied whenever at least one sensor in the room is active. Based on actual occupant behavior, a room is considered occupied whenever there is at least one occupant in that room. The Relative Error between sensed and actual occupancy of each room is given by  $\frac{O_{true} - O_{sensed}}{O_{true}}$ , where  $O_{sensed}$  is the percentage of time the room is occupied according to the sensor network and  $O_{true}$  is the percentage of the time the room is actually occupied.

All simulation runs were performed using DesignDEVS, a



**Figure 6:** Error Plot for Room 15, the hallway.

simulation tool we created for developing and testing DEVS models. The results of our experiment for Room 15, which is a hallway, can be seen in Figure 6. In the plot, observe that as the density of sensors is increased, the error rate decreases in a very smooth trend. However, note that as sensor density goes to 100%, the Relative Error does not go to zero. To understand the reason behind this, see Figure 5f, using Figure 5a to identify Room 15. There are a few areas of the hallway (outside of Room 8, 9, and 10) which are not covered by sensors. These unsensed areas explain the discrepancy.

The distinction between a hallway and the rooms is the fact that rooms have *Waypoints* where occupants stop to perform actions, while a hallway is simply a circulation area where occupants pass through to get to the rooms. Since *Waypoints* may fall outside of the sensed area, the whole time an occupant spends at that location would count towards increasing the Relative Error between sensed and actual occupancy. The resulting error for the rooms can be seen in Figure 7. As an example, see the error plot for Room 5 where the error stays high until 25% density, and then plunges nearly to 0 at 40% density. To understand what happened, see Figure 5b, using Figure 5a to identify Room 5. Observe that the *Waypoint* (a purple dot) is outside of the sensor's range. As we increase the density to 25% (Figure 5c) the *Waypoint* continues to be outside of the sensed area, and finally at 40% density (Figure 5d) we can see the *Waypoint* being inside the sensing area of one of the sensors. Similar relationships between the Relative Error plots and sensor coverage can be seen in other rooms. A slight loss of sensor coverage in an important area can result in a significant error in occupancy measurements. Although the model is simplified, the variability of the plots in Figure 7 is a good reflection of the complication of sensing occupants in real buildings.



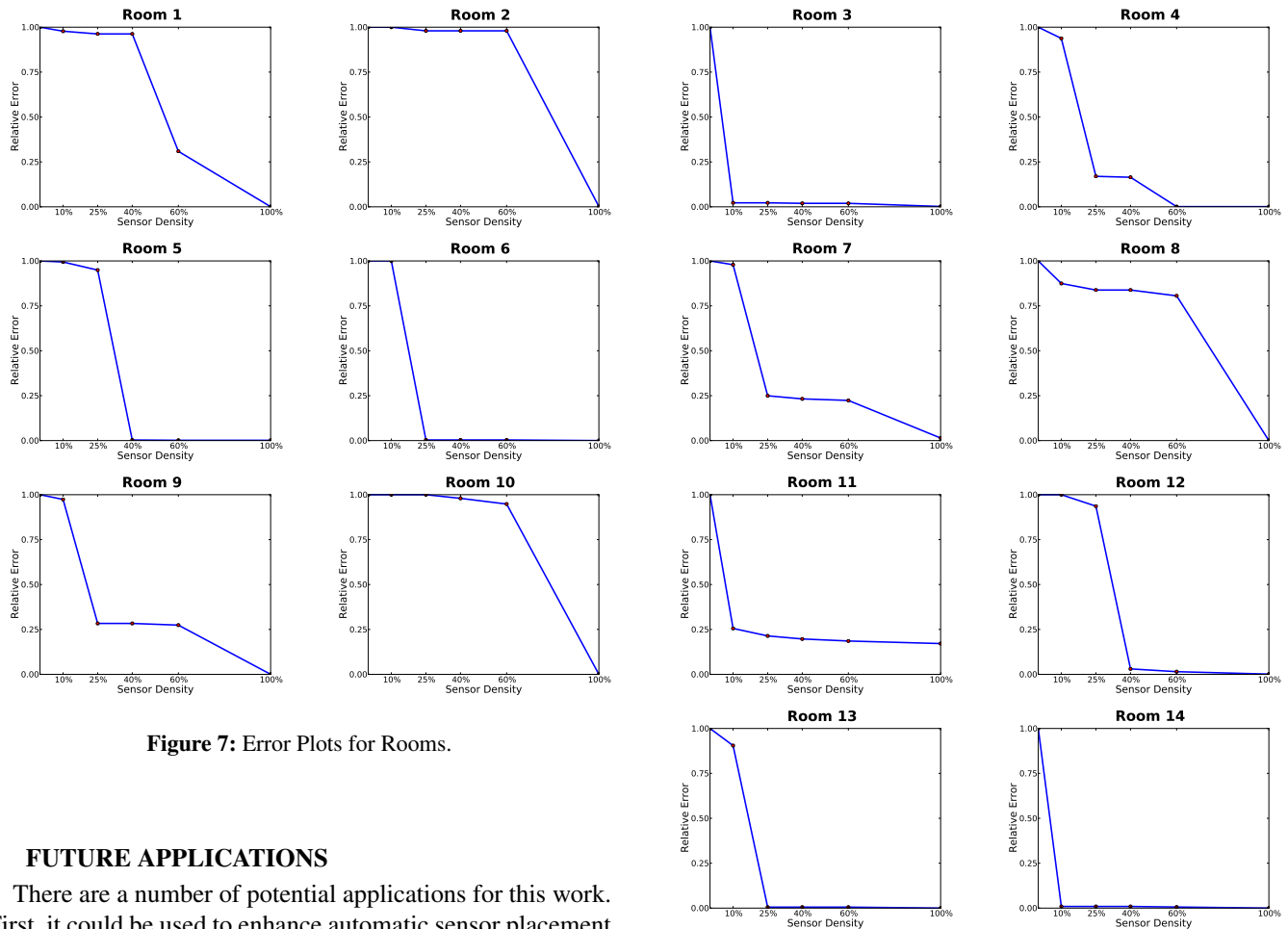


Figure 7: Error Plots for Rooms.

## 5 FUTURE APPLICATIONS

There are a number of potential applications for this work. First, it could be used to enhance automatic sensor placement tools to account for dynamic occupant behavior. Second, with additional models for occupancy sensing technologies based on pressure and ultrasound, the work could support the design of accurate and cost efficient occupant monitoring systems. Simulation models of building occupants and sensors could be integrated with thermal, lighting, and air flow models to improve energy use predictions. They could also support emergency evacuation experiments. To get to the point where such applications are possible, we anticipate a number of necessary enhancements.

For some of these applications, our current occupant behavior is not detailed enough, and new aspects of behavior need to be modeled. Since all our models adhere to the DEVS convention, it is straightforward to replace one model for another. For example, we can easily replace our *Location Selection* model with work by Goldstein et al. (2011), which would help make location selection favor nearby rooms, or steer the occupants away from rooms that are overcrowded. As the occupancy model evolves to account for more environmental factors, we envision omitting both waypoints and the explicit distinction between rooms to start examining emergent occupant behavior. This would introduce a need to dy-

namically adjust navigation paths, requiring enhancements to the *Path Finder* model. For example, we could use a geometrical navigation mesh (Doherty et al., 2012) instead of the grid based navigation to allow for greater freedom of movement. Also, it is easy to replace an existing atomic model with a more complicated hierarchical one. For example, instead of using a simple *Goal Memory* model for the memory, we can adopt a more advanced hierarchical model like Atkinson and Shiffrin (1968), which would have three sub-models: short-term, long-term, and sensory memory.

To expand the simulation to different domains, such as adding thermal and energy models, DEVS allows us to easily add new models adjacent to existing models and connect them using links. For example, for modeling energy consumption we might want to add an *Appliances*, an *HVAC* system, and a *Lighting* model that connects to our *Space Layout* model. Similarly, if we want to expand the occupancy model to support perception of the environment, interaction with the environment, and decision making, we can add new models inside

of the occupancy model, and easily adjust the way submodels are connected. Connecting all these different pieces into one simulation allows us to ask new questions about correlations between different aspects of our environment and occupant behavior.

## 6 CONCLUSION

In this paper we have contributed a hierarchical model of occupant behavior inspired by cognitive research and an experiment that studies how sensor density influences effectiveness of occupancy detection. We have found that the effectiveness of sensors increases as the density of sensors increase, with hallways having a predictable Relative Error, while other room's Relative Error being more variable. By adhering to the DEVS conventions, we have established a foundation for extending our simulation model to support more advanced human behaviors and set a path for future research in our understanding of human behavior in the context of building performance. This research path may help in creating more intelligent Building Automation Systems and increasing designers' understanding of human behavior in buildings.

## References

- Amanatides, J. and A. Woo (1987). A Fast Voxel Traversal Algorithm for Ray Tracing. In *In Eurographics '87*, pp. 3–10.
- Atkinson, R. and R. Shiffrin (1968). Human memory: A proposed system and its control processes. *The psychology of learning and motivation* 2, 89–195.
- Barros, F. J. (1995). Dynamic structure discrete event system specification: a new formalism for dynamic structure modeling and simulation. In *Proceedings of the 27th conference on Winter simulation, WSC '95*, Washington, DC, USA, pp. 781–785. IEEE Computer Society.
- Becker, E., G. Guerra-Filho, and F. Makedon (2009). Automatic sensor placement in a 3d volume. *Proceedings of the 2nd*.
- Bourgeois, D., C. F. Reinhart, and I. A. Macdonald (2005). Assessing the total energy impact occupant behavioural response to manual and automated lighting systems.
- Chiou, Y. (2009). *An Occupant-Driven Integrative Household System For Residential Building Energy Simulation*. Ph. D. thesis, Carnegie Mellon University.
- Doherty, B., D. Rumery, B. Barnes, and B. Zhou (2012). A spatial query & analysis tool for architects. In *Proceedings of the 2012 Symposium on Simulation for Architecture and Urban Design, SimAUD '12*.
- Garaas, T. W. (2011). Sensor placement tool for rapid development of video sensor layouts. In *Proceedings of the 2011 Symposium on Simulation for Architecture and Urban Design, SimAUD '11*, San Diego, CA, USA, pp. 134–137. Society for Computer Simulation International.
- Goldstein, R., A. Tessier, and A. Khan (2011). Space layout in occupant behavior simulation. *IBPSA-AIRAH Building Simulation*, 1073–1080.
- Hart, P., N. Nilsson, and B. Raphael (1968). A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics* 4(2), 100–107.
- Hengel, A., R. Hill, and B. Ward (2009). Automatic camera placement for large scale surveillance networks. *Workshop on Applications of Computer Vision (WACV)*.
- Hoes, P., J. L. Hensen, M. G. Loomans, B. de Vries, and D. Bourgeois (2009). User behavior in whole building simulation. *Energy and Buildings* 41(3), 1–22.
- Narahara, T. (2007). The space re-actor: Walking a synthetic man through architectural space. Master's thesis, Massachusetts Institute of Technology.
- Pelechano, N., J. Allbeck, and N. I. Badler (2008). *Virtual Crowds: Methods, Simulation, and Control*. Morgan & Claypool Publishers.
- US EIA (2008). *Assumptions to the Annual Energy Outlook*. U.S. Energy Information Administration, Washington, D.C.
- Vernon, D., G. Metta, and G. Sandini (2007). A survey of artificial cognitive systems: Implications for the autonomous development of mental capabilities in computational agents. *Evolutionary Computation*.
- Yu, Q. and D. Terzopoulos (2007). A decision network framework for the behavioral animation of virtual humans. *Symposium on Computer Animation*.
- Zeigler, B. P., H. Praehofer, and T. G. Kim (2000). *Theory of Modeling and Simulation* (2nd ed.). San Diego, CA, USA: Academic Press.