Carbon Dioxide as Proxy for Infection

Layeghi, Mahsa
System and Computer Engineering
Carleton University
mahsalayeghi@cmail.carleton.ca

Dandan, Ghida
System and Computer Engineering
Carleton University
ghida.dandan@cmail.carleton.ca

Abstract—**The DEVS formalism is widely used in modeling and simulating different systems. Cell-DEVS is a combination of DEVS and Cellular Automata with explicit timing delays. This paper is concerned with modeling carbon dioxide as proxy for infection using Cadmium Cell-DEVS tool which is used for Discrete-Event modeling and simulation. A lab environment is simulated with occupants under the assumption of no SARS-CoV-2 in the air at the start of the event, and the effect of some parameters on the number of susceptible persons in the room is studied. Such parameters include outdoor air ventilation and the size of the room. The simulation results obtained from varying these parameters are then visualized using DEVS WebViewer, which is a Web visualization tool. The goal is to verify the impact of increasing outdoor air ventilation on minimizing the number of susceptible persons, and to explore other potential solutions for decontaminating the air.**

I. Introduction

COVID-19 is a widespread virus with a high transmission and mortality rate. Solutions to this virus, such as lockdowns, are not feasible in the long run. The notion of reopening public places should be taken with precaution as it runs the risk of yet another outbreak.

According to Peng and Jimenez 2020, COVID-19 transmissions occur via aerosols, i.e., particles or droplets suspended in ambient air (Prather, Wang, and Schooley 2020). These particles or droplets may result from an infected individual's exhalation, for instance. The transmission of aerosols is higher indoors than outdoors, hence imposing health risks on all students or workers who spend more time indoors. It is therefore compulsory to set up counter-measures that reduce indoor transmission of the SARS-CoV-2 virus.

According to current research efforts (Peng and Jimenez 2020), indoor excess CO2 typically manifests itself due to human exhalation, thereby increasing the risk of infection of COVID-19. Outdoor-air ventilation systems, be it natural or mechanical, are a fundamental part of every building design, as they ensure maintaining the indoor air quality. Therefore, the availability of these ventilation systems can be exploited to potentially reduce the risk of infection among faculty and students alike in a lab environment. Unfortunately, further studies show that outdoor-air ventilation systems have limited ability to decontaminate indoor air, thus giving rise to the need for a supplementary air cleaning system (Peng and Jimenez 2020).

In this paper, a model is simulated under different conditions in order to determine the effectiveness of outdoor-air ventilation systems. The results are used to dictate whether an additional air purifying unit would be necessary for different lab conditions. If so, it calculates the optimal equipment specifications for each scenario. This study is conducted using the CADMIUM Cell-DEVS tool, which offers an effective platform for modeling and simulation. DEVS formalism (Gon, Zeigler, and Praehofer 2000) is a simulation technique that has been widely applied to simulate discrete-event dynamic systems.

In the following sections, DEVS and Cell-DEVS are introduced first, followed by a description of the model at hand. The model is simulated and the simulation results are visualized. The visualization results are used to determine whether or not the vents are sufficient independently of air purifying units.

II. Background

The model is developed by Hoda Khalil in Cell-DEVS CD++ and implemented in Cadmium-cell-DEVS by Cristina Ruiz Martin. A Cell-DEVS is a combination of DEVS and cellular automata (CA) with explicit timing delays (Gon, Zeigler, and Praehofer 2000). DEVS was created for modeling and simulating discrete-event dynamic systems (DEDS). It provides a formal framework that supports a full range of dynamic system representation capability, as well as separates modeling from simulation.

A system can be modeled using DEVS as a composition of atomic and coupled components. The atomic model is specified as

$$M =< X, Y, S, \delta_{int}, \delta_{ext}, \lambda, ta >$$

Where X is the set of input events, Y is the set of outputs, S is the set of states, δ_{int} is the internal transition function, δ_{ext} is the external transition function, λ is the output function, and ta is the time advance function.

In figure 1, an informal depiction of the DEVS atomic model can be seen. A DEVS model is in a specific state at each point in time. If there is no external event, the DEVS model stays in that state until the time advance function is passed. After this step, λ is executed and an output Y is generated. On the one hand, if an internal transition is triggered, the state will change. On the other hand, an external transition is triggered by an external event, leading to δ_{ext} determining the new state.

A DEVS coupled model is composed of several atomic or coupled submodules, and it is specified as

$$CO2\_M =< X, Y, M, EIC, EOC, IC, select >$$

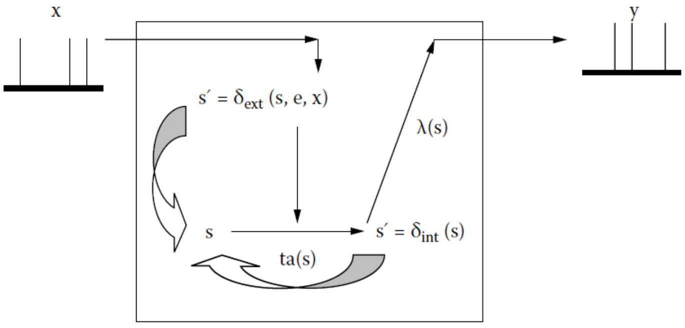

Figure 1: DEVS atomic model

Where X is the set of input events, Y is the set of outputs, M is the set of all component models in DEVS, EIC is the set of external input coupling, EOC is the set of external output coupling, and IC is the set of internal couplings.

An example of a DEVS coupled model is shown in figure 2. It consists of three atomic sub-components; however, combining atomic and coupled sub-components in a coupled model is also possible.

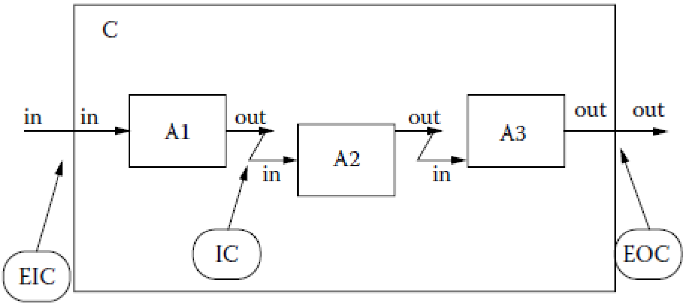

Figure 2: A coupled model

According to G. A. Wainer and Giambiasi 2001, the Cell-DEVS formalism allows for the definition of cell spaces based on DEVS and CA models. In Cell-DEVS, each cell is defined as an atomic model .It can also be integrated into a coupled model representing the cell space.

Figure 3 illustrates an informal description of an atomic cell with transport delays. Upon the arrival of an external event, the local computing function, or τ, is executed, taking N as an input. The result of the local computing function will be transmitted only when there is a state change. Here state change from s to s' is done after a delay of d time units.

Only transparent delay is shown here, but there exist inertial delays with preemptive semantics with no memory. The scheduled events can be discarded if the computed value is different from the future state.

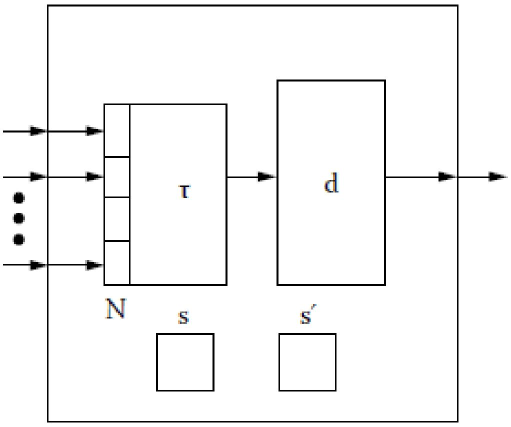

Figure 3: Informal description of an atomic cell with transport delays

After defining the behavior of a single cell, a cell space needs to be formed. A coupled Cell-DEVS can be created by putting together a number of cells interconnected with its neighbors. An example of coupled Cell-DEVS is shown in figure 4. A coupled Cell-DEVS is composed of an array of atomic cells, with given size and dimensions, and each cell is connected to its neighborhood through input/output ports.

CD++ is a modeling and simulation environment developed in C++ following the DEVS and Cell-DEVS' formal specifications and is used to build and execute DEVS and Cell-DEVS models. (G. Wainer 2002). Cadmium is used as a tool for Discrete-Event modeling and simulation, based on the DEVS formalism. The reason behind using Cadmium is to replace CD++ with a more flexible implementation that promises increased performance. Cadmium is a cross-platform header-only library implemented in C++.

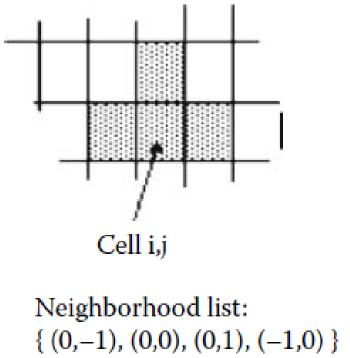

Figure 4: Coupled Model - Cell-DEVS

Using Cadmium (Belloli et al. 2019) for Cell-DEVS models, any item related to the model specification can be specified in a separate JSON file and does not need to be included in the model itself. Such items include the size, borders, and the type and range of the neighborhood. The computing function is defined using a set of rules that results in a state change, and then after a specific amount of delay, the state change will occur.

III. Model Progression

A. *Formal Model Specification*

In this paper, CO2 as proxy for infection is modeled through the use of DEVS and Cell-DEVS. The model is a two-dimensional Cell-DEVS space and it is specified as follows:

$$M =< X_{list}, Y_{list}, S, X, Y, \eta, N, \{t_1, t_2\}, C, B, Z >$$

Type and concentration are the two states defined in this model and are specified on the atomic model. Other parameters like η (neighborhood size), N (neighborhood set), C (cell space), and B (set of border sets) are determined in a JSON file. The model discussed in this paper is unwrapped and is using transport delay as its default delay. The neighborhood is von_neumann of range 1, meaning only North (N), East (E), West (W), and South (S) neighbors are considered in the calculations.

B. *Model Description*

The model at hand is that of the spread of carbon dioxide within a 2D computer lab environment. The computer lab consists of walls (impermeable objects), stationary occupants (CO2 sources), etc. All lab components can be seen in Table 1. During the simulation, workstations turn into a CO2 source once an occupant arrives. Each occupant has a different arrival time, meaning each workstation turns into a CO2 source at a different time during the simulation. Occupants also have a predefined active time (set in the JSON file), after which a CO2 source cell switches back to being a workstation. For the purposes of this paper, the active time is set to last until the end of the simulation run-time, i.e., workstations that switch to CO2 sources remain like that until the simulation ends. These values, as well as other scenario configuration values, can be found in figure 5.

Table 1: The cell types and their corresponding type numbers and colors used in the Python converter.

Cell Type	Type Number	Color	Decimal Color Code (R,G,B)
AIR	-100	White	(255,255,255)
CO2_SOURCE	-200	Cyan	(0,255,255)
IMPERMEABLE_STRUCTURE	-300	Black	(255,255,255)
DOOR	-400	Green	(0,255,0)
WINDOW	-500	Yellow	(255,255,0)
VENTILATION	-600	Blue	(0,0,255)
WORKSTATION	-700	Red	(255,0,0)

```
"default_config": {
    "CO2_cell": {

        "co2_production": 0.0155,
        "cell_size": 25,
        "base" : 500,
        "resp_time" : 1,
        "window_conc": 400,
        "vent_conc": 400,
        "breathing_rate": 8,
        "time_active": 500,
        "start_time": 25,
        "risky_concentration": 600,
        "flow_weight": 0.8,
        "risky_exposure_time": 40,
        "airflow_dir_x": 0,
        "airflow_dir_y": 1
    }
}
```

Figure 5: The configuration of CO2 cells in the .JSON

Each cell type (be it wall, workstation, etc) is defined within the model code by a number as per Table 1. The model requires an input JSON file that describes the lab scenario using the appropriate cell type numbers. Such a lab scenario JSON file can be obtained through the use of a bitmap floor-plan converter tool written in Python, and such a bitmap floor-plan can be seen in figure 6.

Each pixel in the image is 25 × 25 cm, and the tool takes the color of each pixel in the image and corresponds it to a specific cell type, as seen in Table 1. By using the cell type values along with the provided bitmap image, the floor-plan converter outputs a comprehensive JSON file describing the type of each cell, as per the code snippet shown in figure 7.

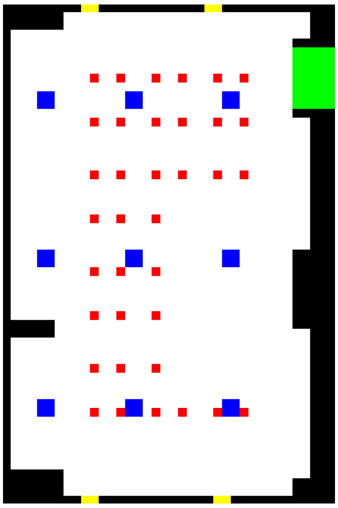

Figure 6: An example input floorplan image

This snippet is that of the resulting JSON file from running a bitmap floor-plan through the Python converter tool. The cell is identified through its coordinates, and its type is derived from the color of the pixel at those coordinates in the bitmap floor-plan. Based on this type, the cell is then assigned a CO2 concentration and a start time (counter). It is important to note that there's a z-coordinate due to the Python converter tool that yields results at all heights within the floor-plan. The ceiling height is assumed to be 13 cells, or 3.25 m (10 ft 8 in). This paper focuses on the height of 4 cells (equivalent to 100 cm) since this is where workstations tops and occupants are present.

As previously mentioned, the `local_computation` function is in charge of changing the state after a delay of d time units has passed, where d is *resp_time* in this case. Figure 8 shows a snippet of code of the `local_computation` function from this model, and is taken from the atomic Cell-DEVS model.

```
    {
        "cell_id": [
            2,
            6,
            4
        ],

        "state": {
            "concentration": 500,
            "type": -700,
            "counter": 9
        }
```

Figure 7: Code snippet from the lab scenario .JSON

In this code snippet, a cell of type `CO2_SOURCE` is shown, and the concentration and number of neighbors values are initialized to 0. The aim is to calculate the CO2 concentration in this specific cell using its neighbors. The code iterates through the neighbors set, which is specified in the JSON file as von_neumann. If a neighboring cell is not of type wall, then it is counted as a neighbor. The new concentration of the `CO2_source` type cell is computed as the sum of its neighboring cells' concentrations divided by the total number of neighbors. Each CO2 source cell type also emits its own CO2 concentration derived from `breathing_counter`, which adds a value to the CO2 concentration every 5 seconds.

If this concentration continues to increase to surpass the `risky_concentration`, it starts checking the `exposure_time`. Should this value exceed the defined `risky_exposure_time`, the CO2 source cell changes its state into a susceptible CO2 source cell. Additionally, at each time step, the function checks if the counter has reached the defined `time_active`, at which point the student is considered to have left the lab, meaning the CO2 source cell would return to being a workstation cell.

C. Running the Simulation

Running the model requires two inputs: the lab scenario JSON file and a time-step value. If a time-step value is not defined, it is assigned 180 steps by default, equivalent to two and a half hours. After running the simulation, the resulting output messages text file and lab scenario JSON file are uploaded to DEVS WebViewer to visualize the results. The normalized simulation files provided by the WebViewer are downloaded. These converted files consist of an options JSON file, a structure JSON file, and a messages log file. Two styles are then integrated within the options JSON file for better visualization purposes, and the three files are re-uploaded onto DEVS WebViewer to obtain a more comprehensible result. At this point, the preferred z-coordinate can be specified within the options JSON file, and this parameter can be changed at any point if the need for it arises.

D. Understanding the Visualization Results

DEVS WebViewer is an online Web application tool that allows for the visualization of Cadmium Cell-DEVS simulation results. This model's simulation results contain three ports: the counter port, the CO2 concentration port, and the type port, as seen from left to right in figure 9. It is important to note that the color mapping in the visualization differs from that described in earlier sections, which is only adapted in the Python floor-plan converter.

1) Counter Port: The leftmost rectangle in figure 9 represents the counter port, which is the time at which each occupant arrives into the room. Each occupant is represented by a blue cell. It serves the purpose of viewing the different arrival times for the occupants based on their counters. Since each occupant acts as a CO2 source, whenever the counter for each workstation reaches its start time (defined in the JSON file), it switches to a CO2 source. Likewise, after a specific amount of time, each CO2 source switches back to a workstation, which signalizes that the occupant's active time is up (i.e., the student has left the lab). The active time is set until the end of the simulation runtime by default in the JSON file.

2) CO2 Concentration Port: The second rectangle in figure 9 maps the concentration of CO2 in the room (shown as a red gas), as well as it shows the clean outdoor air pumped into the room from either vents or windows (shown as a blueish gas). The CO2 concentration of each cell is calculated by the sum of the CO2 concentration of the neighboring cells divided by the total number of neighbors. This concentration gradually increases the longer an occupant exists. When the CO2 concentration exceeds a certain threshold around a workstation, it is considered as risky concentration. The high concentration of CO2 can be clearly distinguished from the outdoor air towards the upper side of the room, for instance.

```
/** Model developed by Hoda Khalil in Cell-DEVS CD++
* Implemented in Cadmium-cell-DEVS by Cristina Ruiz Martin */
case CO2_SOURCE:{
    int concentration = 0;
    int num_neighbors = 0;
    for(auto neighbors: state.neighbors_state) {
        if( neighbors.second.concentration < 0){
            assert(false && "co2 concentration cannot be negative");
        }
        if(neighbors.second.type != IMPERMEABLE_STRUCTURE){
            concentration += neighbors.second.concentration;
            num_neighbors +=1;
        }
    }
    new_state.concentration = (concentration/num_neighbors);
    if( (new_state.breathing_counter % breathing_rate ) == 0){
        new_state.concentration += concentration_increase;
        }
    new_state.breathing_counter++;
    new_state.counter += 1;
    if (state.current_state.counter == time_active ) {
        new_state.type = WORKSTATION;
    }
    else
    if (new_state.concentration >= risky_concentration){
        new_state.exposure_time++;
        if (new_state.exposure_time >= risky_exposure_time) {
            new_state.type = SUSCEPTIBLE_CO2_SOURCE;
        }
    }
    break;
}
```

Figure 8: CO2 source cell type from the atomic model

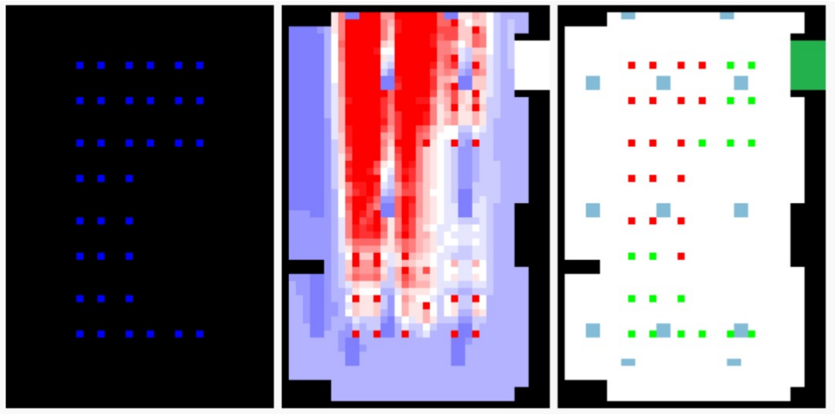

Figure 9: Example of visualization results. The ports are, from left to right: counter port, CO2 concentration port, type port

3) Type Port: The rightmost rectangle in figure 9 represents each cell type, be it a workstation, wall, CO2 source, or susceptible individual. It aids in showing where most susceptible persons are based on how high the CO2 concentration is. When a healthy occupant is exposed to risky concentration for a specific amount of time (risky exposure time) or more, the occupant becomes susceptible to infection, shown as a red cell in this rectangle.

IV. Simulation Parameters

The simulation results allow for the study of multiple cases, such as the rate at which occupants become susceptible, how long the occupants are susceptible for on average, the number of susceptible occupants in different time ranges, etc.

In this paper, the first susceptible occurrence was the primary focus, along with the total number of susceptible occupants within a range of 10 time-steps starting from the time of first susceptible until the end of the simulation. It's important to note that a susceptible occupant switches back to a healthy one if he does not remain within risky concentration for a certain amount of time.

Several simulation scenarios have been constructed in order to find the effect of specific parameters on the values, as mentioned earlier. Such parameters include the room size, the number of vents, and the number of windows.

Two different room sizes were explored, the default one (135 m^2 or 1457 sqft) and a smaller one (84 m^2 or 308 sqft). For each room size, the number of vents was varied between zero, three, and nine, and for each vent count, the scenario either had zero or four windows, as summarized in Table 2.

The windows were assumed to either be all closed, or all opened, simultaneously. Realistically, as per fluid dynamics, the number of open windows, as well as their positioning, play a major role on the airflow in the space. This fact was passed over due to the team's lack of experience in fluid mechanics.

Observing the change in rate of increase of CO2 concentration in smaller environments is crucial since according to Peng and Jimenez 2020, the rate of change in CO2 is inversely proportional to volume. This means that in a smaller room with same ventilation conditions, it is expected to see more susceptible occupants.

Additionally, the simulation was also performed on the same set of floor-plans but with a vent concentration of 200 ppm instead of 400 ppm. These scenarios were simulated for visualisation purposes only, to see whether the rate at which the CO2 concentration in the room reaches the risky concentration (600 ppm) would be lower if the vent CO2 concentration was lowered. The results, however, were insignificant and the vent concentration will not be further elaborated on.

Table 2: Summary of the different lab scenarios with their sizes, number of vents, and number of windows.

Scenario #	Size (cells)	Area (sqm)	# Vents	# Windows
1	38 × 57	135.375	0	0
2	38 × 57	135.375	3	0
3	38 × 57	135.375	9	0
4	38 × 57	135.375	0	4
5	38 × 57	135.375	3	4
6	38 × 57	135.375	9	4
7	27 × 50	84.375	0	0
8	27 × 50	84.375	3	0
9	27 × 50	84.375	9	0
11	27 × 50	84.375	0	4
11	27 × 50	84.375	3	4
12	27 × 50	84.375	9	4

V. Results

A. Increasing Outdoor Air Ventilation

Outdoor air ventilation can be achieved either by adding vents or windows to the floor-plan. The number of vents and windows combinations discussed in Table 2 are simulated, and the simulation results are visualized. Visualizing the simulation results allows for a better understanding of how the change in parameters affects the scenario. Figure 10 showcases how the increase of vents gradually decreases the CO2 concentration in the cells, or in other words, an increase in air changes per hour (ACH).

Upon adding four windows to each of the aforementioned vent number cases, it becomes clear that windows play a significant role in increasing the ACH of the room, as per figure 11. All three scenarios in this figure contain four windows, while the number of vents is varying; however, the three results look quite similar, with no clear distinction among them, as was seen earlier in figure 10.

At time-step 100, the number of susceptible people is counted in intervals of 10 time-steps (Table 3), then the numbers are compared for each time interval among different scenarios. The results show that the number of susceptible persons decreases significantly as the ventilation measures are increased; however, the number does not quite reach a negligible value. Peng and Jimenez 2020 state that the increase in ventilation would eventually minimally affect the change in CO2 concentration of a room. This brings to light the possibility for a different mechanism of air cleaning.

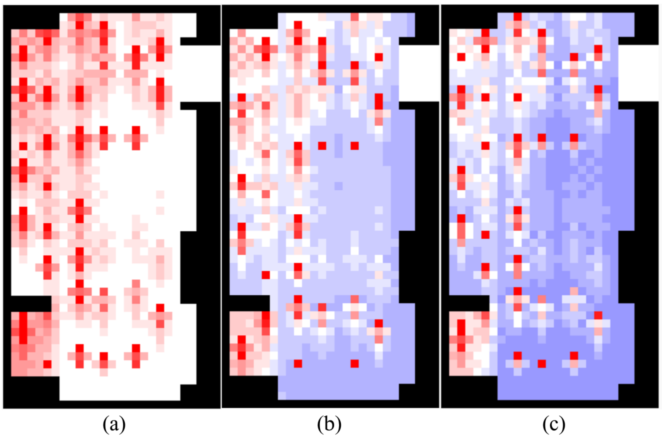

Figure 10: Visualization of simulation results with no windows and (a) no vents, (b) three vents, and (c) nine vents.

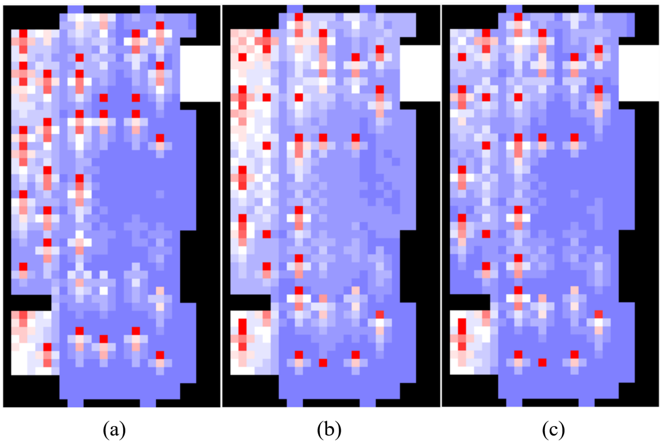

Figure 11: Visualization of simulation results with four windows and (a) no vents, (b) three vents, and (c) nine vents.

B. *Room Size Manipulation*

The room size is decreased from the default one while keeping the number of occupants the same but at different number of vents. The number of windows for these simulations was kept static at zero due to the results shown previously where the effect of windows overtook vents.

In figure 12, it can be seen that in the smaller scenario, there appear to be more areas with elevated CO2 concentrations than there are in the larger scenario. Both scenarios have poor ventilation conditions and the same number of occupants; however, in a smaller environment, it is expected to see higher concentrations of CO2 as opposed to a more spacious environment. These results are confirmed by Peng and Jimenez 2020 who state that the concentration of CO2 increases more rapidly in smaller spaces.

In Figure 13, the number of vents is increased to three in both areas. The vents are evenly spaced and in a column on the right side of the lab. Figure 13 shows how the smaller room becomes more densely populated with relatively cleaner air, specifically on the right side of the room where the vents are located, as opposed to the larger room.

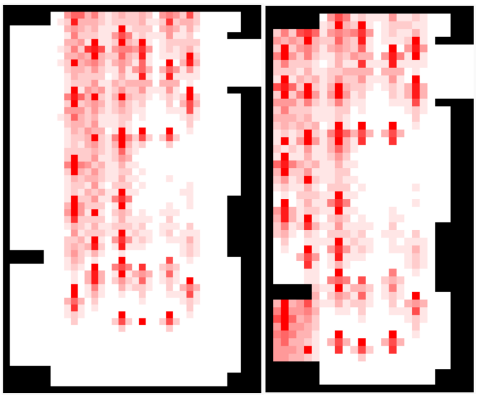

Figure 12: Zero vents and windows in a lab of 375 sqm (left) and 85 sqm (right)

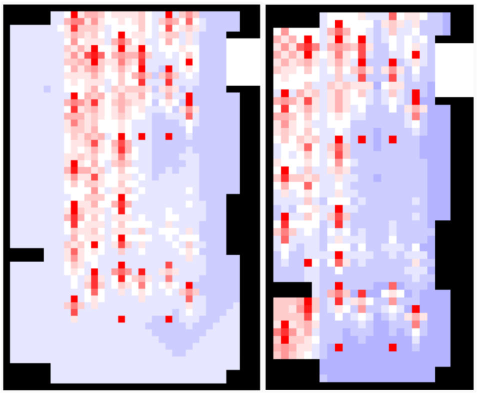

Figure 13: Three vents and zero windows in a lab of 375 sqm (left) and 85 sqm (right)

Table 3: The number of susceptible persons within the two time intervals Δt_1 from 130 to 140 time-steps and Δt_2 from 150 to 160 time-steps in each lab scenario, where A,V,W represents the area (in sqm), number of vents, and number of windows.

Scenario	A,V,W	# Susceptibles Δt_1	# Susceptibles Δt_2
1	[135,0,0]	13	10
2	[135,3,0]	11	4
3	[135,9,0]	10	4
4	[135,0,4]	3	4
5	[135,3,4]	3	4
6	[135,9,4]	3	4
7	[84,0,0]	15	10
8	[84,3,0]	9	7
9	[84,9,0]	10	7
11	[84,0,4]	3	5
11	[84,3,4]	3	5
12	[84,9,4]	2	5

Although the smaller room size affects the rate at which healthy occupants turn susceptible, the introduction of an air cleaning unit counters this disadvantage since smaller rooms typically require less clean air delivery rate (CADR) according to Shaughnessy and Sextro 2006.

C. *Effects on First Susceptible Occurrence*

As previously mentioned, the effect of certain parameters on the rate of increase in carbon dioxide concentration is to be studied. The simulation results are viewed using DEVS WebViewer, and the instance at which a healthy occupant turns susceptible for the first time is noted as per Table 4. To reiterate, a healthy occupant turns susceptible once exposed to the predefined risky concentration for a risky amount of time.

Crucial information concerning the rate of spread of CO2 can be derived from comparing the first occurrence of a susceptible individual: if this event occurs sooner in one scenario over the other, this means that the risky concentration was reached sooner, therefore faster, than it did in the other scenario. A higher value for first susceptible occurrence time is thus favorable over a lower one.

Table 4: The first occurrence of a susceptible individual for each lab scenario represented by t_{first} expressed in time-steps, where A,V,W represents the area (in sqm), number of vents, and number of windows.

Scenario #	A,V,W	t_{first} (in time-steps)
1	[135,0,0]	99
2	[135,3,0]	106
3	[135,9,0]	111
4	[135,0,4]	116
5	[135,3,4]	122
6	[135,9,4]	121
7	[84,0,0]	96
8	[84,3,0]	109
9	[84,9,0]	111
11	[84,0,4]	114
11	[84,3,4]	115
12	[84,9,4]	115

A commonality among the obtained results showed that windows provide significantly more outdoor air ventilation than vents do. In terms of first susceptible occurrence, figure 14 offers a comparative view on how the addition of windows delays the exposure of healthy occupants to risky concentrations of CO2, as opposed to using vents independently. While the increase of the number of vents from 0, to 3, then 9, increases the time of first susceptible occurrence by a total of 12 time-steps, the addition of 4 windows further pushes back this time by another 10 time-steps. It is important to note, however, that this increase is not indefinite: the increase of outdoor ventilation may only assist to a certain limit, according to Peng and Jimenez 2020. This is actually confirmed by the simulation result visualization, since the time of first susceptible occurrence eventually ceases to increase, as per figure 15.

VI. Inclusion of Air Purifier Units

According to Li et al. 2007, there is a connection between the transmission rate of airborne infectious diseases and ventilation. Additionally, Jones et al. 2020 state that supplementing outdoor air ventilation with portable air cleaners equipped with HEPA filters further assists in eliminating malicious particles.

According to a spreadsheet calculator built to support the report written by Jones et al. 2020, it is optimal to have 5 air changes per hour (ACH), where

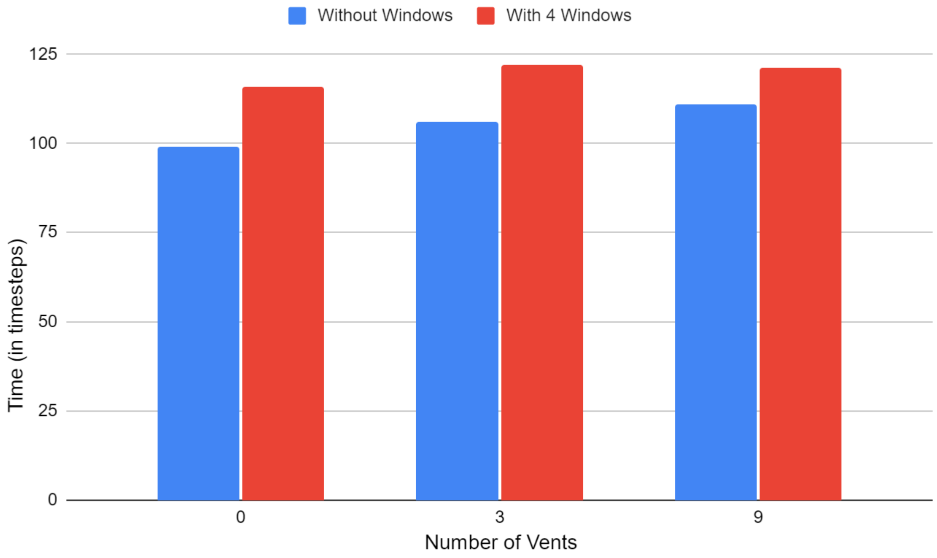

Figure 14: Column chart showing the time (in time-steps) at which a healthy occupant first turns susceptible for different numbers of vents, with and without windows.

each air change would exchange the indoor air (contaminated) with outdoor air (clean). This exchange occurs as a result of ventilating the room through the use of vents or windows. While the goal is to achieve 5 ACH, outdoor ventilation system rates are typically around 3 ACH in learning institutions, according to ASHRAE 62.1 (2019) standard. The inclusion of a portable air purifier in a room with preexisting outdoor ventilation would increase the total ACH, which is equivalent to the sum of the outdoor ventilation ACH and the portable air cleaner ACH. In addition to increasing ACH, portable air cleaners also aid in filtering out airborne respiratory droplets containing the SARS-CoV-2 virus according to Morawska et al. 2020.

Given the different lab scenarios explored in this paper, calculations were performed to determine the necessary clear air delivery rate (CADR) to be provided by the air cleaner unit. This is due to the fact that a poorer outdoor ventilated lab would require an air purifier unit with a higher CADR than a better ventilated one. Similarly, a smaller room would require less CADR from the air cleaning device in comparison to a larger space (Shaughnessy and Sextro 2006).

In order to calculate the required CADR given the lab scenario, each scenario's ventilation setting is taken into consideration. The previously mentioned spreadsheet calculator is used to determine the ACH of the rooms based on how well ventilated they are. The range of ACH values is given as per Table 5.

The ACH values disclosed in Table 5 were adapted to the lab scenarios previously discussed (Table 6) to effectively estimate the necessary CADR that the air purifying unit should have.

$$CADR = \frac{V \times \delta_{ac}}{60}$$

Where V is the volume of the room in m^3, and δ_{ac} is the air cleaner's ACH in h^{-1} according to Shaughnessy and Sextro 2006. Since the ACH of the room is additive, meaning the ACH from ventilation and ACH from air cleaning unit are summed to achieve the total ACH in a room, then the required ACH of the air cleaner, or δ_{ac}, can be calculated by subtracting the outdoor air ventilation ACH from the optimal total ACH, i.e.:

$$\delta_{ac} = ACH_{desired} - ACH_{VentSetting}$$

The resulting CADR values can be seen in Table 7. These CADR values provide valuable insight on what type of air purifying units different lab environments with differing outdoor ventilation systems or size can benefit from. While air cleaners with larger CADR values do exist, larger laboratories that require such an elevated CADR value can exploit the additive property of ACH, and instead implement the usage of multiple air cleaning units with lower CADR instead of a singular one.

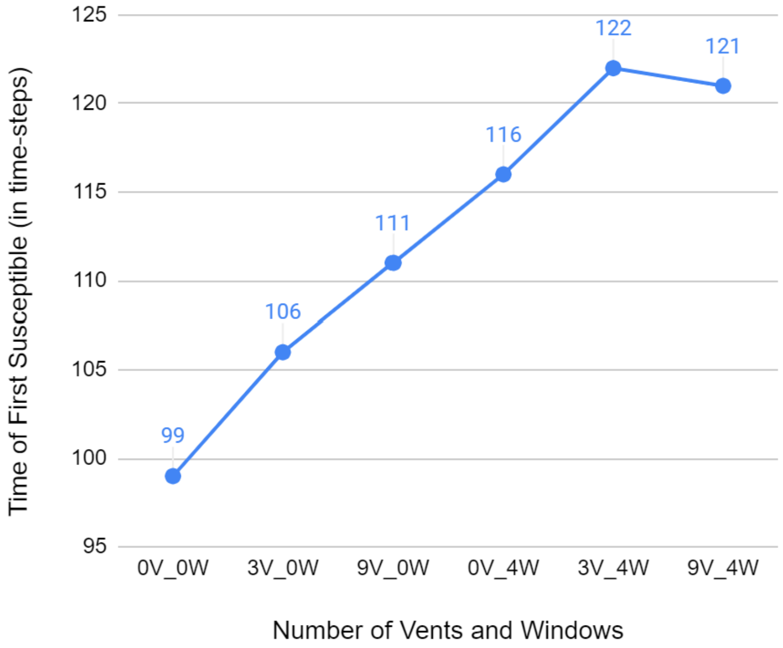

Figure 15: Line chart showcasing the change in first susceptible occupant occurrence (in time-steps) as a function of increase of outdoor ventilation

Table 5: ACH values corresponding to different ventilation settings. The “No Ventilation” setting is added outside the scope of the spreadsheet calculator for the purposes of this paper.

Ventilation Setting	ACH
Enhanced Ventilation	4
Good Ventilation	3
Typical Ventilation	1.5
Low Ventilation	1
No Ventilation	0.5

Table 6: The ACH corresponding to the different lab scenarios, where A,V,W represent the area in sqm, the number of vents, and the number of windows respectively.

Scenario	A,V,W	ACH
1	[135,0,0]	0.5
2	[135,3,0]	1
3	[135,9,0]	1.5
4	[135,0,4]	3
5	[135,3,4]	4
6	[135,9,4]	4
7	[84,0,0]	0.5
8	[84,3,0]	1
9	[84,9,0]	1.5
10	[84,0,4]	3
11	[84,3,4]	4
12	[84,9,4]	4

Table 7: Recommended air cleaning unit CADR values based on the ventilation settings and the lab sizes.

Scenario	A,V,W	CADR (m^3/min)	CADR (cfm)
1	[135,0,0]	33	1155
2	[135,3,0]	29	1027
3	[135,9,0]	26	898
4	[135,0,4]	15	513
5	[135,3,4]	7	257
6	[135,9,4]	7	257
7	[84,0,0]	21	720
8	[84,3,0]	18	640
9	[84,9,0]	16	560
10	[84,0,4]	9	320
11	[84,3,4]	5	160
12	[84,9,4]	5	160

VII. Conclusions

The model simulated using Cadmium Cell-DEVS yielded significant results in terms of outdoor-air ventilation effectiveness and its impact on different room sizes. The increase in outdoor-air ventilation decreases the rate at which CO2 concentration increases, subsequently lowering the total number of susceptible occupants present in a lab environment in the long run. It is important to note that occupants remained in this environment for the equivalent of around 5 to 6 hours; however, no occupant remained susceptible until the end of their active time. Although this is favorable, it's preferred to further minimize the number of susceptibles, as they could get infected while they are in the environment nevertheless. Unfortunately, outdoor-air ventilation systems are unable to provide optimal air decontamination, which gives rise to the necessity of another air purifying unit such as portable air cleaners. These units are classified by clean air delivery rate, and different enclosed spaces of varying size or outdoor-air ventilation systems require different variations of the CADR value.

Acknowledgments

The simulations performed for this paper were made possible thanks to the efforts of ARSLab members. The model used was developed by Hoda Khalil and Cristina Ruiz Martin. ARSLab, or Advanced Real-Time Simulation Laboratory is supervised by Prof. Gabriel A. Wainer, and is located in Carleton University, Ottawa, Ontario, Canada. This paper also makes use of "CELL-DEVS Models for CO2 Sensors Locations in Closed Spaces" written by Hoda Khalil, Gabriel Wainer, and Zachary Dunnigan.

References

Belloli, Laouen et al. (2019). "Building DEVS models with the cadmium tool". In: *2019 Winter Simulation Conference (WSC)*. IEEE, pp. 45–59.

Gon, Kim Tag, Bernard P Zeigler, and Herbert Praehofer (2000). "Theory of modeling and simulation: integrating discrete event and continuous complex dynamic systems". In:

Jones, Emily et al. (2020). "Healthy schools: risk reduction strategies for reopening schools". In: *Harvard TH Chan School of Public Health Healthy Buildings program.*

Li, Yiping et al. (2007). "Role of ventilation in airborne transmission of infectious agents in the built environment-a multidisciplinary systematic review." In: *Indoor air* 17.1, pp. 2–18.

Morawska, Lidia et al. (2020). "How can airborne transmission of COVID-19 indoors be minimised?" In: *Environment international* 142, p. 105832.

Peng, Zhe and Jose L Jimenez (2020). "Exhaled CO2 as COVID-19 infection risk proxy for different indoor environments and activities". In: *medRxiv.*

Prather, Kimberly A, Chia C Wang, and Robert T Schooley (2020). "Reducing transmission of SARS-CoV-2". In: *Science.*

Shaughnessy, RJ and RG Sextro (2006). "What is an effective portable air cleaning device? A review". In: *Journal of Occupational and Environmental Hygiene* 3.4, pp. 169–181.

Wainer, G (2002). "CD++: a toolkit to define discrete-event models". In: *Software, Practice and Experience* 32.3, pp. 1261–1306.

Wainer, Gabriel A and Norbert Giambiasi (2001). "Application of the Cell-DEVS paradigm for cell spaces modelling and simulation". In: *Simulation* 76.1, pp. 22–39.