CADMIUM CELL-DEVS MODEL OF CO_2 AS A PROXY FOR COVID INFECTION WITHIN A DYNAMIC SETTING

SUMMARY

Term Project

SYSC 5104

Author: Sebastien Cook

101022643

Date Submitted: December 20th, 2020

Table of Contents

1.0 Introduction

COVID-19 has become an ever-present worry in people's daily lives. Many guidelines have been put in place to reduce the risk of transmission and protect the population. For example, people are required to sanitise their hands before entering public locations to reduce contact transmission; wearing a mask has been enforced as a law in public locations since masks block the transmission of most airborne contagion droplets produced by talking, coughing, sneezing. Furthermore, the two-meter distancing rule has been put in place to limit airborne transmission since COVID-19 is largely transmitted via large droplets which are too heavy to float, dropping to the ground within two meters unless coughing or sneezing, in which case the large droplets can reach distances of up to six meters [1]. However, despite the multiple countermeasures that have been set in place, there is still a significant threat of transmission via the lesser produced small droplets that are light enough to float in the air for 15 minutes up to three hours [2]. Although the quantity of small droplets exhaled is very small, over time they accumulate in badly vented locations and can pose a serious threat to any occupant's health. This threat is especially prevalent in indoor settings since simply circulating the air with a fan will not remove the contagion; it requires ventilation to replace the air [2].

By combining and modifying "CO_2 Model with Moving Occupants" with "Model of CO_2 as a Proxy for Infection", this paper seeks to simulate the risk of COVID-19 transmission via aerosolized contagion in a dynamic indoor setting by simulating and analysing the concentration of CO_2 throughout.

2.0 Background

This project uses Cell-DEVS formalism to design and build the models. Cell-DEVS is a simulation using cellular automata, where each cell is a DEVS model that is connected to its neighbours.

DEVS offers a formalized modeling technique that functions in a modular, hierarchical manner. At its base, DEVS provides the framework for atomic models which can be as simple as a random number generator. This can then be coupled with other atomic models, creating coupled models. These coupled models can further be coupled, hence the modular hierarchical nature of the formalism. This leads to a very reusable and expandable modeling method since atomic models can be added, remover or modified individually. Furthermore, DEVS is easily validated since each aspect can be tested and validated individually or combined.

The atomic model is defined as an input, output, states, internal transition, external transition, output function and time advance (see Figure 1). Notably, an atomic model does not necessarily require any inputs. For example, it could be a variable generator for another atomic model. The state variable(s) are associated with the time advance function which determines the amount of time a model spends in a given state. When the time of the given state ends, the internal transition and output function are triggered. The internal transition function will change or remain in the state depending on its state variables. The output function will propagate the result of the model's current state on the output ports. Finally, when an input arrives at the input port(s) X, an external transition δ_{ext} is triggered. It is the implementation of δ_{ext} that determines what to do with the received information. The flow of a DEVS atomic model can be seen in Figure 2 below.

$$\text{DEVS atomic model} = < X, Y, S, \delta_{ext}, \delta_{int}, \lambda, ta>$$

Figure 1. DEVS formal definition [3]. X is the input port(s) and Y is the output port(s) of the model. S is the state variable(s) of the model which is associated with the time advance function ta. δ_{int} and λ represent the internal transition and output function, respectively. The external transition is represented by δ_{ext}.

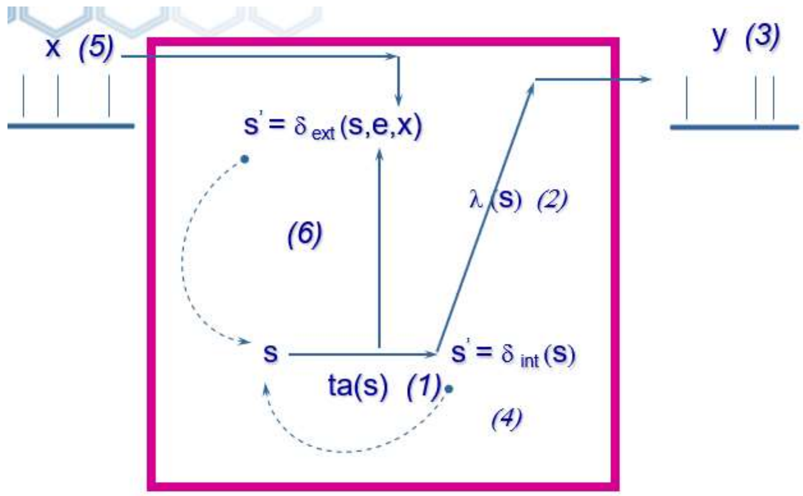

Figure 2. DEVS atomic model transition flow

Cellular automata is a simulation technique comprised of a grid of cells that each have their own state and implementation. Each cell communicates with its assigned neighbours. When a cell's transport delay (dt) is expended, it broadcasts its state information to its neighbours. The neighbours, upon receiving the input, will trigger their input computing function (τ). There is also an internal delay (di) which is pre-emptive and meant to maintain the models timing despite interruptions such as receiving an input.

Cell-DEVS is the combination of cellular automata with DEVS formalism. Each cell is an atomic model that are coupled, creating a grid of atomic models which, together, comprise the Cell-DEVS model. For this paper, the Cadmium Cell-DEVS environment was used. Cadmium offers many different DEVS and Cell-DEVS features. Most importantly, it offers the use of multiple state variables within each cell atomic model. Cadmium is implemented in C++ and is built in a hierarchical manner of classes that each implement the functions of the model.

3.0 Base Model Background

Before discussing the model that was designed and implemented in this paper, understanding the base models upon which the model of interest was built is paramount. "CO_2 Model with Moving Occupants" and "Model of CO_2 as a Proxy for Infection", two models implemented in Cadmium Cell-DEVS, were combined and further expanded. They are both based on the original "Cell-Devs Models for CO_2 Sensors Locations in Closed Spaces" (CO_2CS), implemented in CD++ [4], which modeled the dissipation of CO_2 in a closed environment to find the optimal location for CO_2 sensors to detect occupation. The two base models used for the expanded implementation, used the core principles of CO_2 dissipation rules and ventilation established in CO_2CS, implemented them in Cadmium Cell-DEVS and further added their own layers of complexity. This was possible due to the modular nature of Cell-DEVS. Both base models are part of the Simulation Everywhere – Cell DEVS CO_2 spread indoors GitHub [5].

3.1 CO_2 Model with Moving Occupants

As mentioned above, this model is based on the original CO_2CS. However, this model adds an extra layer of complexity by adding a dynamic CO_2 generator. In the original CO_2CS model, the CO_2 generator was static, spawning at its assigned location at the beginning of the simulation, and remaining there, producing CO_2, until the simulation ended. However, in "CO_2 Model with Moving Occupants", the CO_2 generators spawn at a predetermined location like a door, and travel around the setting.

"CO_2 Model with Moving Occupants" has 3 state variables: concentration, type, and count. The type state variable is associated with transport delay, defined as: 5 seconds if type is a CO_2 producer, and one otherwise; and concentration. A Vonnumen neighbourhood is used, meaning that the neighbourhood is comprised of itself and the 4 surrounding pols (ie. North, South, East, and West). Furthermore, the model has 3 lists: actionList, studentList, workstationList. These lists save the next cell to be traveled to, all of the CO_2 generator information and locations, and the CO2 generator assigned workstation and its location, respectively. The behaviour of each individual cell depends on their state variables. Below in table one is pseudo code describing a cells behaviour when it receives and input depending on its "type".

Table 1: CO_2 Model with Moving Occupants τ pseudocode

T(N) concentration	N = type
Concentration = average neighbourhood CO_2 concentration If any action list position = this cell position → type = CO_2 source If spawn point and maximum students isn't reached → spawn a new student at the entrance every two seconds	Air: -100
Concentration = average neighbourhood concentration + 12.16ppm Determine next action If do nothing → do nothing If move → type = Air and add action to action list	CO_2_source: -200
Concentration = 0	Wall: -300
Concentration = 500	Door: -400
Concentration = 400	Window: -500
Concentration = 300	Vent: -600
Concentration = average neighbourhood CO_2 concentration	Workstation: -700

This model was tested in a setting representing a computer lab that contained ventilation along the walls, multiple rows of desks, and a big door. The simulation had multiple students (i.e. CO_2 sources) spawn at the door, travel to their assigned workstation, wait a random amount of time, and then exit by the entrance. As the students traveled to their assigned workstations, a trail of CO_2 lingered behind. This trail was the minimal concentration. However, with many students entering the setting, the CO_2 concentration quickly climbed to very high levels.

The dynamic feature of the CO_2 sources means that the CO_2 spread through the room much faster than it would have if the CO_2 generators were static. Furthermore, since CO_2 is slow to dissipate, it quickly increased in concentration, reaching alarmingly high levels.

3.2 Model of CO_2 as a Proxy for Infection

"Model of CO_2 as a Proxy for Infection" is also built upon the CO_2CS model. However, it retains little of the original model. Although maintaining static CO_2 sources, it adds an extra layer of complexity to the dissipation of CO_2 and further introduces a new "type" state variable: susceptible CO_2 source. These changes serve to simulate the CO_2 sources that become at risk should CO_2 carry the COVID-19 contagion inside a room with an air current.

Table 2: Model of CO_2 as a Proxy for Infection τ pseudocode

T(N) concentration	N = type
If X_wind == neighbor relative position[X] and Y_wind == neighbor relative position[Y] → flow concentration = (neighbors' concentration*wind) Else → concentration += neighbors' concentration If flow concentration != 0 → Concentration = (1-wind)*concentration/(num_neighbors-1) + flow concentration Else → Concentration = average neighbourhood concentration If flow concentration != 0 → concentration = (1 -1 flow weight)*average concentration of all but upwind neighbour + flow concentration Else → concentration = average neighbourhood concentration	Air: -100
If X_wind == neighbor relative position[X] and Y_wind == neighbor relative position[Y] → flow concentration = (neighbors' concentration*wind) Else → concentration += neighbors' concentration	CO_2 source: -200

If flow concentration != 0 -> Concentration = (1-wind)*concentration/(num_neighbors-1) + flow concentration + 12.16ppm Else → Concentration = average neighbourhood concentration + 12.16ppm If in threshold quantity of CO_2 → count++ If exposure time = risky time → type = susceptible CO_2 source and add next action to susceptible action list If count = time active -> type = workstation	
Concentration = 0	Wall: -300
Concentration = 500	Door: -400
Concentration = 400	Window: -500
Concentration = 300	Vent: -600
If X_wind == neighbor relative position[X] and Y_wind == neighbor relative position[Y] → flow concentration = (neighbors' concentration*wind) Else → concentration += neighbors' concentration If flow concentration != 0 -> Concentration = (1-wind)*concentration/(num_neighbors-1) + flow concentration Else → Concentration = average neighbourhood concentration If start time → type = CO_2 source	Workstation: -700
If X_wind == neighbor relative position[X] and Y_wind == neighbor relative position[Y] → flow concentration = (neighbors' concentration*wind) Else → concentration += neighbors' concentration If flow concentration != 0 → Concentration = (1-wind)*concentration/(num_neighbors-1) + flow concentration Else → Concentration = average neighbourhood concentration If concentration < risky concentration → type = CO_2 source and exposure time = 0	Susceptible CO_2 source: -800

If count = time active → type = workstation	

The first major change to the model is the addition of an air current. The air current is implemented by adding two new configuration values, X and Y, which denote the direction of the air current. For example, if the air current values are X = 1 and Y = 0, then there is a very strong wind in the north direction only. This is translated in CO_2 dissipation by increasing the average concentration in the same direction as the wind.

A subsequent addition to the model which can be seen in table two is the new "type" state variable: susceptible CO_2 source, which has a type value of -800. A regular CO_2 source can become susceptible if they remain in a threshold concentration of CO_2 for a predetermined amount of time. Once susceptible, the susceptible CO_2 source then stops producing CO_2. However, they can return to being a regular CO_2 source if the CO_2 concentration drops below the susceptible threshold. After a time active, both regular and susceptible CO_2 sources will de-spawn at which point, the simulation ends.

This model was tested using the same computer lab setting as "CO_2 Model with Moving Occupants" used. It further configured an air current value of X = 1 and Y = 0. The simulation demonstrated that the CO_2 sources downwind became susceptible the fastest since the CO_2 from all other sources accumulated downwind. Eventually, all CO_2 sources became susceptible. However, all the CO_2 sources upwind became susceptible from their own CO_2 dissipation.

4.0 Model Description

The model designed in this paper was built by combining and modifying the two models described above. This process was completed in three steps, such that the models could each be tested and validated before the addition of a new feature. The first step was combining both models in their simplest forms to create a dynamic susceptible model. This model used the susceptible type presented by "Model of CO_2 as a Proxy for Infection" along with the dynamic CO_2 sources implemented in "CO_2 Model with Moving Occupants". After successfully validating the dynamic susceptible model, a dynamic infected model was implemented. This model further introduced a new "type" state variable: infected CO_2 source, seeking to simulate the risk of becoming susceptible in a setting with a confirmed COVID-19 infected person. Finally, a dynamic infected complex model was produced. This model reintroduced the air current from the "CO_2 as a Proxy for Infection" model along with a susceptible risk based on distance of the source. This is the final form of the model presented in this paper; it seeks to simulate the risk of becoming susceptible if a COVID-19 infected person is introduced to the closed setting in a more realistic way than its predecessor models.

4.1 Dynamic Susceptible Model

The dynamic susceptible model is the first iteration. It was built by combining both base models in their simplest forms, using the dynamic implementation of the CO_2 sources along with the extra "type" state variable: susceptible CO_2 source. However, in this model, when a CO_2 source becomes susceptible, they continue producing CO_2 and can never return to the regular CO_2 source type.

Table 3: Dynamic Susceptible Model τ pseudocode

T(N) concentration	N = type
Concentration = average neighbourhood CO_2 concentration If any action list position = this cell position → type = CO_2 source If any susceptible action list position = this cell position → type = susceptible CO_2 source If spawn point and maximum students isn't reached → spawn a new student at the entrance every two seconds	Air: -100
Concentration = average neighbourhood concentration + 12.16ppm Determine next action If do nothing → do nothing If move → type = Air and add action to action list If in threshold quantity of CO_2 → count++ If count = risky time → type = susceptible CO_2 source and add next action to susceptible action list	CO_2 source: -200
Concentration = 0	Wall: -300
Concentration = 500	Door: -400
Concentration = 400	Window: -500
Concentration = 300	Vent: -600
Concentration = average neighbourhood CO_2 concentration	Workstation: -700
Concentration = average neighbourhood CO_2 concentration + 12.16ppm Determine next action If do nothing → do nothing If move → type = Air and add action to susceptible action list	Susceptible CO_2 source: -800

As it can be in table three that the air current seen in the "CO_2 as a Proxy for Infection" model is not implemented in this iteration. This is because it was an extra layer of complexity that would have complicated the validation of the combined base model. Another interesting change that can be observed in the pseudo code above is the addition of a susceptible action list. This allows air cells to know whether to become a CO_2 source or a susceptible CO_2 source since the asynchronous nature of Cell-DEVSE means that the CO_2 source may have de-spawned before the air cell could check the surrounding types.

As mentioned above, the dynamic susceptible model is the combination of both base models in their simplest forms. Pulling aspects from both models, the dynamic susceptible model models the worst-case scenario of COVID-19 infection by assuming that every CO_2 generator may transmit COVID-19 through their CO_2 emissions. However, as seen in "CO_2 as a Proxy for Infection", it is possible for a regular CO_2 source remaining at their workstation for an extended period to make themselves into a susceptible CO_2 source. This is a shortcoming of the current model. That being said, it still serves as a good baseline model of the worst-case scenario in an enclosed space.

4.2 Dynamic Infected Model

The dynamic infected model is the second iteration, built off of the model described in Section 4.1. In this iteration, a new type state variable along with modified CO_2 dissipation rules were added. The infected CO_2 source is a third CO_2 source type added to the model, representing a confirmed COVID-19 infected person. The purpose of this model is to simulate the risk of becoming infected should a confirmed COVID-19 infected person be introduced in a closed space. This provides a more realistic risk analysis compared to the worst-case analysis of the dynamic susceptible model.

Table 4: Dynamic Infected Model τ pseudocode

T(N) concentration	N = type
Concentration = average neighbourhood CO_2 concentration If any action list position = this cell position → type = CO_2 source If any susceptible action list position = this cell position → type = susceptible CO_2 source If any susceptible action list position = this cell position → type = infected CO_2 source If spawn point and maximum students isn't reached → If infected maximum isn't reached → spawn a new infected student at the entrance every two seconds Else → spawn a new student at the entrance every two seconds	Air: -100
Concentration = average neighbourhood CO_2 concentration Determine next action If do nothing → do nothing If move → type = Air and add action to action list If in threshold quantity of CO_2 → count++ If count = risky time → type = susceptible CO_2 source and add next action to susceptible action list	CO_2 source: -200
Concentration = 0	Wall: -300
Concentration = 500	Door: -400
Concentration = 400	Window: -500
Concentration = 300	Vent: -600
Concentration = average neighbourhood CO_2 concentration	Workstation: -700

Concentration = average neighbourhood CO_2 concentration Determine next action If do nothing → do nothing If move → type = Air and add action to susceptible action list	Susceptible CO_2 source: -800
Concentration = average neighbourhood CO_2 concentration + 12.16ppm Determine next action If do nothing → do nothing If move → type = Air and add action to infected action list	Infected CO_2 source: -900

Apart from the addition of a new state variable type, the dissipation of CO_2 rules was also changed. This is reflected in table 4, where the infected CO_2 source produces CO_2. All other CO_2 sources simply calculate their concentration like the air type cell does. This allows for the exclusive modelling of CO_2 that can carry COVID-19 as an airborne contagion following the same rules of infection as in the previous iteration. Furthermore, a third action list for the infected CO_2 source was added to allow the air cells to know which type of CO_2 source it should become.

The changes between this iteration and the previous one are minimal. However, the quantity of new information is large. The dynamic susceptible model modeled a worst-case scenario where every source could infect another or even themselves. The dynamic infected model simulates the specific risk of being infected by airborne COVID-19 contagion in a closed space. The dynamic infected model can be used to analyze ventilation and workstation placement to find the optimal solution for minimal infection. However, the model still lacks the complexity of the real world, despite providing a useful simple setting analysis.

4.3 Dynamic Infected Complex Model

The dynamic infected complex model is the final iteration of the model presented in this paper. It is built upon the dynamic infected model, adding two new features to create more realistic model. The previous model only simulated the infection of COVID-19 through small droplet airborne contagion. Small droplets have a lower viral load, meaning a person requires longer exposure to be bombarded with enough contagion. However, at a two-meter proximity to the source, large droplets are possibly present, containing a higher viral load. This means that a person can be exposed for a shorter amount of time before becoming at risk.

In the new iteration, the two-meter rule is implemented. If a regular CO_2 source is within two meters of the infected CO_2 source, the time to become susceptible is reduced significantly. However, since the CO_2 sources are dynamic, a trail of the higher infection cells are left in the infected CO_2 sources wake for thirty seconds, simulating the time frame in which large droplets can remain airborne. The second new feature is the re-addition of airflow within the model. Even in closed spaces, there is often an air current. This is a step towards producing a more realistic and usable model.

Table 5: Dynamic Infected Complex Model τ pseudocode

T(N) concentration	N = type

If X_wind == neighbor relative position[X] and Y_wind == neighbor relative position[Y] → flow concentration = (neighbors' concentration*wind)
 	
Else → concentration += neighbors' concentration
 	
If flow concentration != 0 → Concentration = (1-wind)*concentration/(num_neighbors-1) + flow concentration
 	
Else → Concentration = average neighbourhood CO_2 concentration
 	
If any action list position = this cell position → type = CO_2 source
 	
If any susceptible action list position = this cell position → type = susceptible CO_2 source
 	
If any susceptible action list position = this cell position → type = infected CO_2 source
 	
If spawn point and maximum students isn't reached →
    If infected maximum isn't reached → maybe spawn a new infected student at the entrance every two seconds
    Else → spawn a new student at the entrance every two seconds
 	
If neighbours' risky distance larger than risky distance → risky distance = neighbours' risky distance - 1
 	
If risky distance counter == 30 and risky distance larger than 0 → risky distance – 1 and counter = 0
Else if risky distance larger than 0 → counter +1
	Air: -100

If X_wind == neighbor relative position[X] and Y_wind == neighbor relative position[Y] → flow concentration = (neighbors' concentration*wind) Else → concentration += neighbors' concentration If flow concentration != 0 → Concentration = (1-wind)*concentration/(num_neighbors-1) + flow concentration	CO_2 source: -200

Else → Concentration = average neighbourhood CO_2 concentration
 	
Determine next action
    If do nothing → do nothing
    If move → type = Air and add action to action list
 	
If in threshold quantity of CO_2 →
    If risky distance between 3 and 4 → count += 4
    Else if between 1 and 2 → count += 2
    Else → count += 1
    If count = risky time → type = susceptible CO_2 source and add next action to susceptible action list
 	
If neighbours' risky distance larger than risky distance → risky distance = neighbours' risky distance - 1
 	
If risky distance counter == 30 and risky distance larger than 0 → risky distance – 1 and counter = 0
Else if risky distance larger than 0 → counter +1	

Concentration = 0	Wall: -300
Concentration = 500	Door: -400
Concentration = 400	Window: -500
Concentration = 300	Vent: -600
If X_wind == neighbor relative position[X] and Y_wind == neighbor relative position[Y] → flow concentration = (neighbors' concentration*wind) Else → concentration += neighbors' concentration If flow concentration != 0 → Concentration = (1-wind)*concentration/(num_neighbors-1) + flow concentration Else → Concentration = average neighbourhood concentration If neighbours' risky distance larger than risky distance → risky distance = neighbours' risky distance - 1 If risky distance counter == 30 and risky distance larger than 0 → risky distance – 1 and counter = 0 Else if risky distance larger than 0 → counter +1	Workstation: -700
If X_wind == neighbor relative position[X] and Y_wind == neighbor relative position[Y] → flow concentration = (neighbors' concentration*wind)	Susceptible CO_2 source: -800

Else → concentration += neighbors' concentration If flow concentration != 0 → Concentration = (1-wind)*concentration/(num_neighbors-1) + flow concentration Else → concentration = average neighbourhood CO_2 concentration Determine next action If do nothing → do nothing If move → type = Air and add action to susceptible action list If neighbours' risky distance larger than risky distance → risky distance = neighbours' risky distance - 1 If risky distance counter == 30 and risky distance larger than 0 → risky distance – 1 and counter = 0 Else if risky distance larger than 0 → counter +1	
If X_wind == neighbor relative position[X] and Y_wind == neighbor relative position[Y] → flow concentration = (neighbors' concentration*wind) Else → concentration += neighbors' concentration If flow concentration != 0 → Concentration = (1-wind)*concentration/(num_neighbors-1) + flow concentration + 12.16ppm Else → Concentration = average neighbourhood concentration + 12.16ppm Determine next action If do nothing → do nothing If move → type = Air and add action to susceptible action list If neighbours' risky distance larger than risky distance → risky distance = neighbours' risky distance - 1 If risky distance counter == 30 and risky distance larger than 0 -> risky distance – 1 and counter = 0 Else if risky distance larger than 0 -> counter +1	Infected CO_2 source: -900

The two new features, namely the two-meter rule (risky distance) and the air current, have been added to the cell types: Air, Workstation, CO_2 source, Susceptible CO_2 source and Infected CO_2 source. The wind strength is determined in the configuration json file as a percentage. Although unconventional, it is a simple way of determining the weight each cell will carry and contribute to the average concentration. Since the model uses a Vonnumen neighbourhood, the air current can only come from the North, South, East and West exclusively – to model a crosswind, the model would require a Moors neighbourhood. The neighbouring upwind cell of a cell calculating its CO_2 concentration will be multiplied by the wind modifier, while the all the other neighbouring cells are multiplied by one minus the wind modifier, meaning the upwind cell experiences a larger effect from the cell down wind from its position.

The two-meter rule is the second feature added. If a neighbouring cell has a risk level above zero, the current cell's risk level will be the neighbours risk level minus one since its one farther from the source. After thirty seconds, the risk level is reduced by one until it returns to zero. The maximum risk level is four since the two meters are represented by 4 cells. The Cell an infected CO_2 source passes through becomes a level 4 risk. All bordering cells become a level 3, and so on until the threat level reaches 0. If a CO_2 source is within one meter of an infected source or its path within 30 seconds of their passing, risk level 3 or 4 cell, the CO_2 source will become susceptible four times faster. Within a two-meter distance, meaning a risk level between one and two, the time to infection is doubled comparatively to the common time and half as long as the highest risk level. This serves to simulate the increase in risk proportional to the distance of the infected CO_2 source.

The air current and the two-meter rule are two features that serve to complicate the model's model of infection, allowing it to serve as a more realistic model of infection than the dynamic infection model. However, it is still far from the real system. The iterative method of development allows to test each feature individually and ensure that it will serve as a sturdy base model for future work.

5.0 Experimental Framework and Results

Three subsequent interactions were produced and tested, each before producing the next. This allowed for the testing and validation of each new feature to ensure a strong base for the next iteration to build upon. In this section the tests and results of each model simulation will be discussed in chronological order.

To allow cross analysis, each model was simulated over the same period of 1,000 time units. Each setting was tested using the same settings and configurations when possible. Furthermore, Figure 3 shows the colour legend of the CO_2 concentrations used in every simulation. By ensuring that the models are tested in the most similar fashion possible, previous iterations can be used to validate the behaviour of the next one.

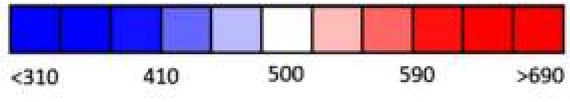

Figure 3. Color legend of CO_2 concentration in ppm [4]

5.1 Dynamic Susceptible Model Simulation

The dynamic susceptible model is the culmination of combining "Model of CO_2 as a Proxy for Infection" and "CO_2 Model with Moving Occupants" in their simplest forms. The purpose was to both create a solid base for the next iteration, but also model the worst-case possibility of becoming infected with COVID-19 from airborne contagion, assuming everyone could transmit it.

Figure 4 illustrated the visualization of the simulation using the computer lab scenario presented in the base models. The computer lab has a large door on the East side, ventilation along the North, West, and South walls, and a series of workstations in the center of the room. CO_2 sources spawn at the door and travel to their randomly assigned workstations, remain for a random amount of time, and then return to the door and de-spawn. A regular CO_2 source is red and becomes purple if it becomes a susceptible CO_2 source type.

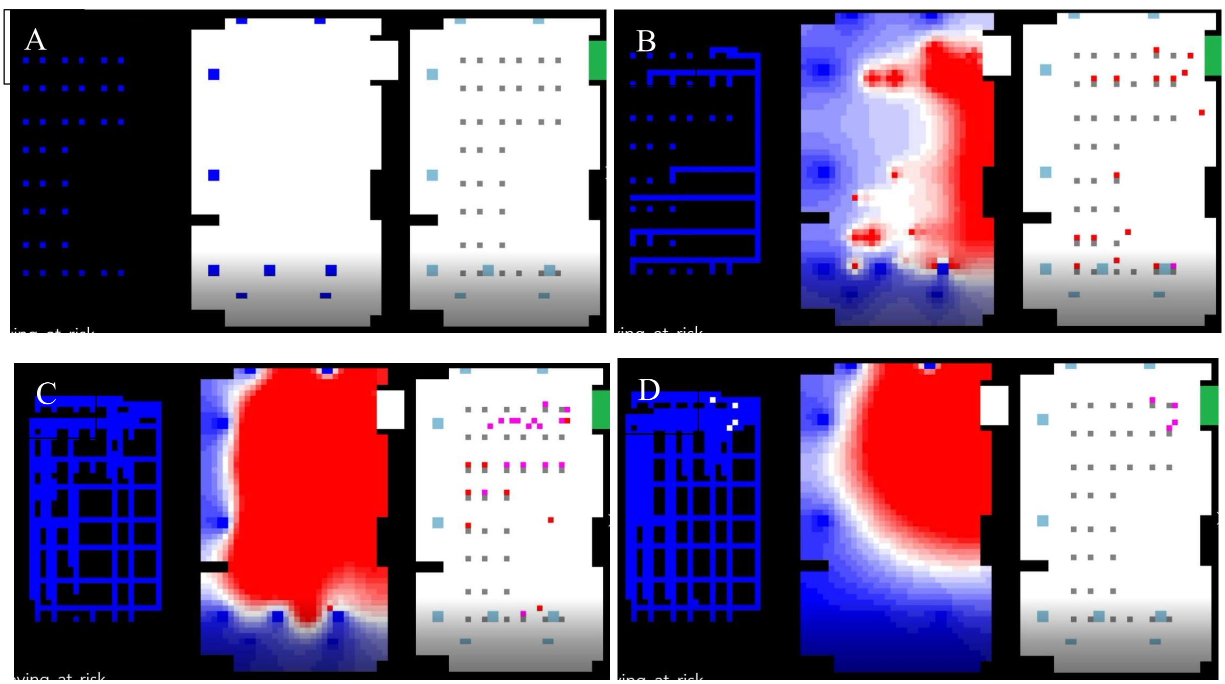

Figure 4. Dynamic susceptible model computer lab simulation. The simulation has 3 visualization windows. The window on the left demonstrates the path taken by CO_2 sources. The window in the middle is the levels of CO_2 in ppm, and the window on the right is the position of the CO_2 sources.

Looking at the visualization it can be determined that the dynamic nature of the CO_2 sources spreads CO_2 quickly along the path of travel as the concentration of CO_2 reaches very high levels. One of the CO_2 sources in the bottom right corner has become susceptible in image B. In image C, all CO_2 sources have spawned, and many are headed back to the exit. From this, it is clear that the ventilation in the computer lab, although numerous, has little effect on the levels of concentration of the CO_2. Despite the lack of presence in the bottom left corner of the workstations, the CO_2 levels remain high. At the end of the simulation, most CO_2 sources have exited the lab, and the ventilation is finally starting to dissipate the CO_2 concentration. Although not demonstrated in the visualization, every CO_2 source in the simulation became susceptible before leaving their workstations.

The first purpose of this simulation visualization is to validate the model. Looking at the visualization, the CO_2 sources reliably spawn at the door until the maximum number of sources is reached. Each CO_2 source travels to their designated workstation and becomes susceptible when exposed to high concentrations. The second point of interest for validation are the susceptible CO_2 source rules. When a regular CO_2 source is exposed to the risky concentration (600 ppm in this simulation) for the risky duration (40 seconds in this simulation), they become susceptible. Once susceptible, they maintain the behaviour of a regular CO_2 source and return to the exit. The behaviour demonstrated in the visualization above is as expected and validates it based on the expected outcome.

The second purpose of the simulation is to conduct a risk analysis of the worst-case scenario. Considering that every CO_2 source in the simulation became susceptible, the risk of becoming susceptible is 100% with this model. This brings to light two important problems. The first is the ventilation in the setting. Since the ventilation is only on three of the four walls, it allows CO_2 to accumulate in concerningly high concentration on the side without ventilation. Ventilation should be placed in the center of the room to further improve its efficiency by dissipating the CO_2 on all four sides of the unit rather than only the side facing the CO_2. The second important problem of the model is the fact that CO_2 sources can infect themselves. Every CO_2 source only became susceptible when at their workstation. This means they likely submitted themselves to a high concentration and the surrounding CO_2 sources only expedited the process (i.e. they did not definitively cause the transition to being susceptible).

The simulation of the dynamic susceptible model demonstrated that the model was successfully combined and produces an acceptable worst-case scenario for airborne infection.

5.2 Dynamic Infected Model Simulation

The dynamic infected model is the second iteration, built upon the dynamic susceptible model after it was validated. The dynamic infected model seeks to model the risk of being infected by COVID-19 if an infected person is introduced to a closed setting. It models the aerosolized contagion, specifically the small droplets contagion traveling with the CO_2 emissions.

Once again, this model was simulated in the computer lab setting. The same amount of CO_2 sources spawned, however two of them were infected CO_2 sources. As described in the model description, only the infected CO_2 source produce CO_2 in the simulation. The other CO_2 sources simply act as air cells, tracking the concentrations they are exposed to and becoming susceptible cells if exposed to too much for too long.

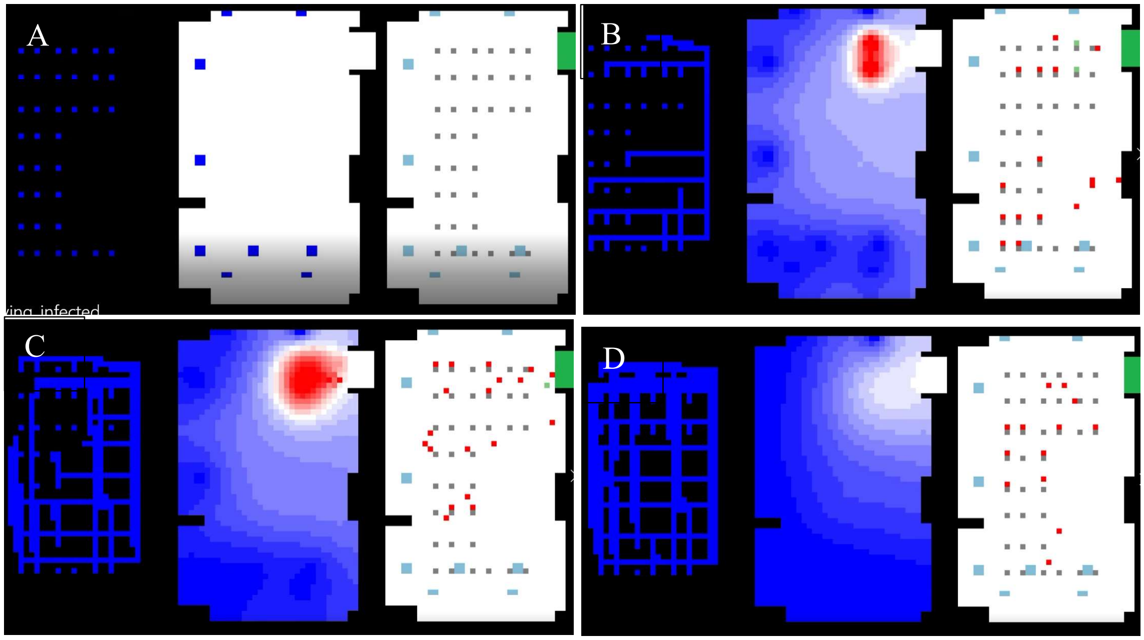

Figure 5. Dynamic infected model computer lab simulation. Infected CO_2 sources are denoted by a light green. When an air cell becomes susceptible, it is denoted in purple.

As in the previous iteration, all CO_2 sources regardless of being regular, susceptible, or infected, have the same behaviour. They spawn at the door, travel to a workstation, return to the door, and de-spawn. Maintaining the configurations across the iteration tests is important for both validation and cross analysis. It can be seen in image B that both infected CO_2 sources have spawned and reached their randomly assigned desks. It is noticeable that the ventilation is keeping the concentration of CO_2 very isolated to the individual in question. In image C, both infected CO_2 sources leave their workstation and are headed for the door. The CO_2 has expanded, but the workstations are spread too far apart for any regular CO_2 source to be exposed to high concentration.

During the simulation, no CO_2 source became susceptible. This is likely because the workstations were too far apart, and the ventilation within the room was adequate to keep emissions from both infected CO_2 sources under control. For this reason, a new setting was introduced to further test the model.

The new setting is a representation of a bus (see Figure 6). The only sources of ventilation present in the bus are a window and the door at the front. CO_2 sources spawn at the door until every seat is filled. They travel to their seat, sit for a random amount of time the return to the door and de-spawn. This simulation is meant to represent a tighter closed environment were distancing is limited. Again, two infected CO_2 sources are introduced to the system.

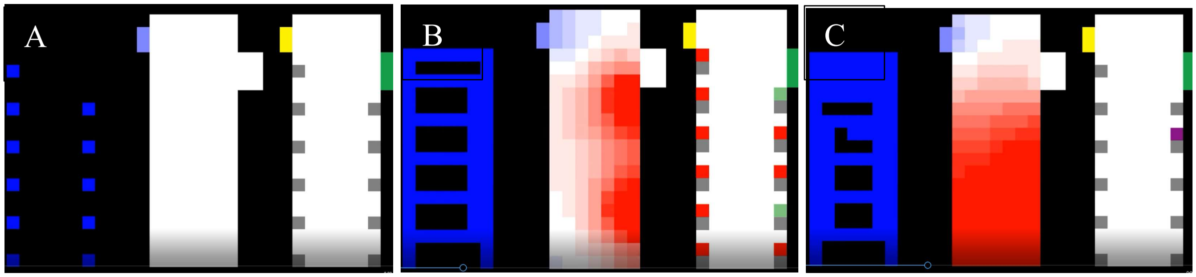

Figure 6. Dynammic infected model bus simulation. Two sources of ventilation are present on the bus, namely a window (yellow) and a door (green). The infected CO_2 sources are depicted in green and the regular CO_2 sources are depicted in red. Once a regular CO_2 source becomes susceptible, it is shown in purple.

The lack of ventilation is noticeable since the CO_2 concentration quickly rises. Both infected sources are sitting on the same side of the bus, however by image C, the entire occupied space is enveloped in their emissions. By the end of the simulation, five regular CO_2 sources – all CO_2 sources in the infected column along with two others in the same row as the infected sources – became susceptible. This demonstrates that proximity plays a large role in COVID-19 transmission since almost all sources in the opposing column remained unsusceptible. The second simulation setting served to validate the functioning of the dynamic infected model, and further highlights the importance of ventilation in closed spaces to circulate the air containing the airborne pathogen.

5.3 Dynamic Infected Complex Model Simulation

The dynamic infected complex model is the final iteration of the model presented in this paper. It seeks to model the risk of being infected by COVID-19 if an infected person is introduced to a closed setting using a more complex set of rules. It models both the small and large droplets exhaled in the CO_2 of the infected CO_2 sources.

This model was tested and simulated in two parts. First, the computer lab configuration with winds of 25% from the North and West was used. Next, the bus configuration with winds of 25% from the North and West and winds of 75% from the North was tested. The increased north wind within the bus configuration simulates the bus driving with an open window at the front, allowing a strong draft wind into the bus. The remaining parameters set in the configuration file are the same as the dynamic infected model simulations. However, a new block on the bottom left of each image was added to track the risk level of each cell. Black is the basic a risk level of zero, yellow is a risk level of one to two, and orange is a risk level of three to four. Finally, for ease of analysis, the infected CO_2 sources were always assigned the same workstations throughout each simulation.

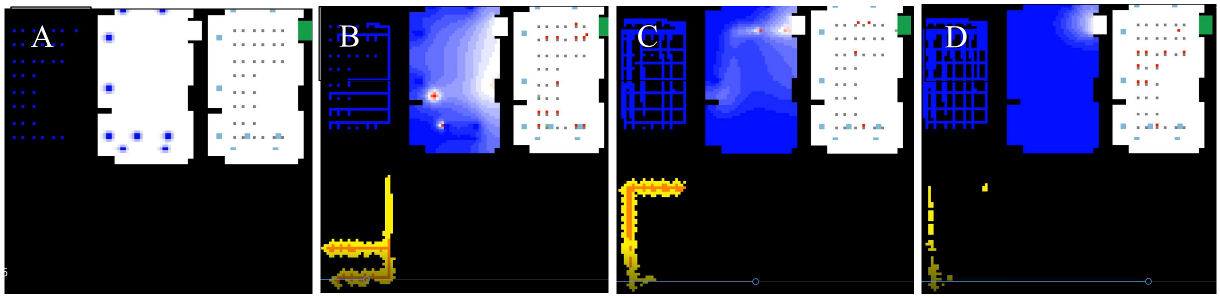

Figure 7. Dynamic infected complex model lab simulation with a West wind of 25%.

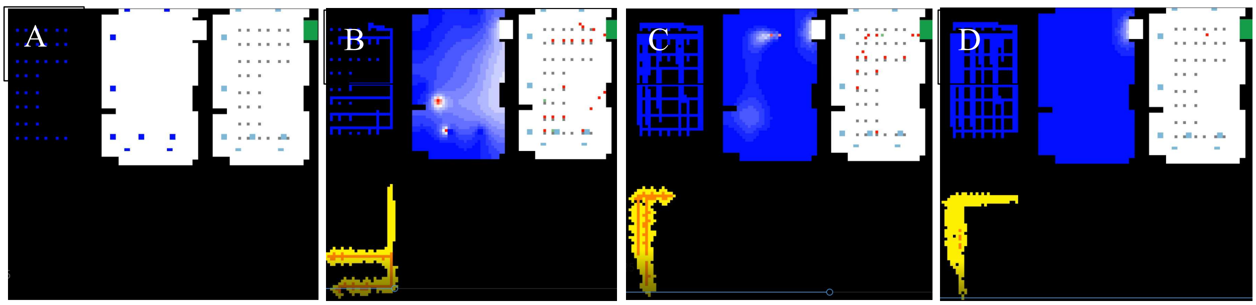

Figure 8. Dynamic infected complex model lab simulation with a North wind of 25%.

Above is the visualisation of the dynamic infected complex model with both a West and North wind, respectively. As mentioned above, the infected CO_2 sources were assigned to the same workstation in both simulations. This leads to a very similar pathing, which explains the near identical cell risk level path.

As in the second iteration test in the lab, no regular CO_2 source becomes susceptible. This is because the air current introduced into the system improved the efficiency of the ventilation by quickly dissipating the CO_2. In the simulation of the dynamic infected model, the concentration of CO_2 in the middle right of the lab is a lighter blue comparatively to the simulation of the dynamic infected complex model. This indicates that the air current allows the ventilation to cycle air in much less ventilated locations of the closed space. Although simply circulating the air using a fan doesn't remove the infection contagion, it allows the clean air from the ventilation to better dissipate the CO_2 and therefor the contagion. This demonstrates the importance of air circulation.

The higher risk cells did not change the outcome of this simulation since the room is likely too large, meaning that CO_2 sources do not come on close enough contact for any significant change. Therefore, the analysis of a more confined space like the bus configuration is important.

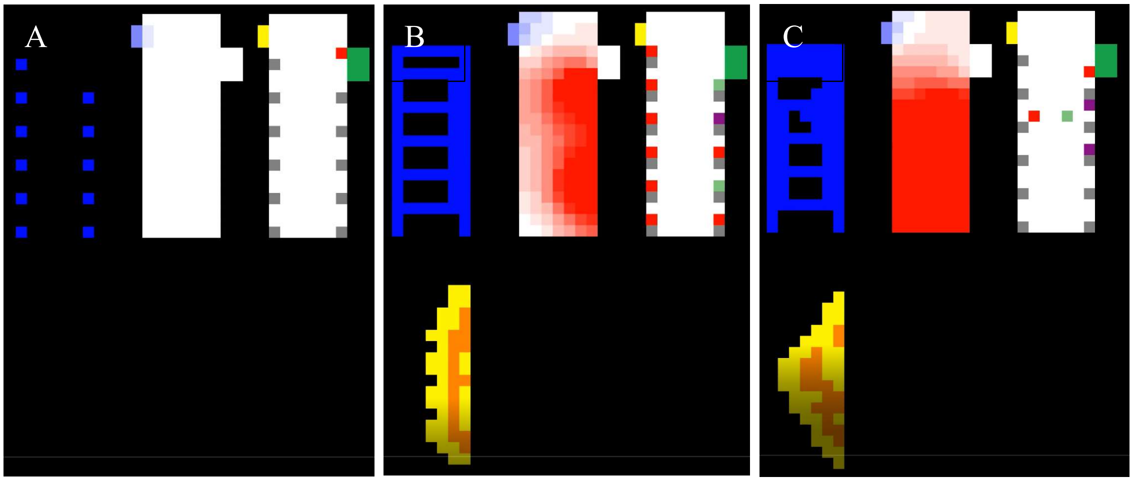

Figure 9. Dynamic infected complex model bus simulation with a West wind of 25%.

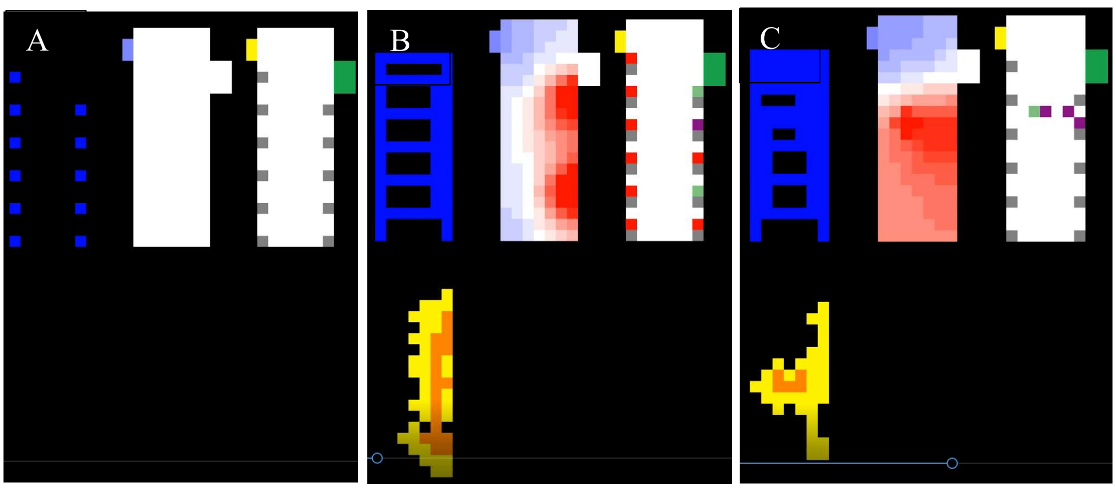

Figure 10. Dynamic infected complex model bus simulation with a North wind of 25%.

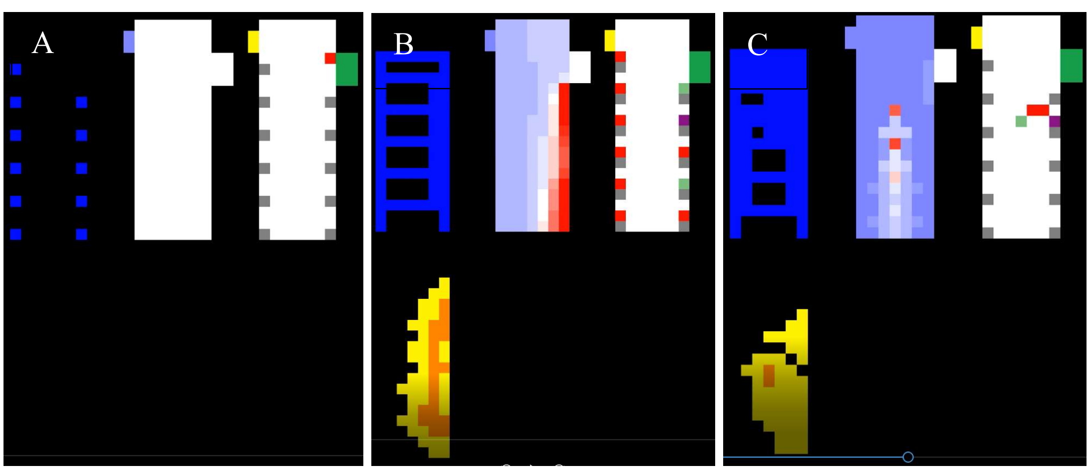

Figure 11. Dynamic infected complex model bus simulation with a North wind of 75%.

The visualization of the dynamic infected bus simulation and the complex bus with a West wind of 25% are very similar. However, the difference is that the wind allowed the CO_2 to spread in the bus much more efficiently. The concentration of CO_2 in previously low concentration areas such as the left side of the bus are much higher when a wind is present, despite blowing in the opposite direction. This demonstrates that with a lack of ventilation, an air current improved rather than inhibited the spread of infected CO_2.

However, this observation was short lived and likely a special circumstance since, in the bus simulation with a North wind of 25%, the concentration of CO_2 remains very controlled. The concentration of CO_2, despite all the other configurations being the same, is much lower than both the dynamic infected model lab simulation and the West wind simulation. This is likely because the wind is pushing the clean lower concentration air from the window and door, down into the bus where it maintains a relatively low infected CO_2 level.

The final simulation completed in this model is with a North wind of 75%. This was intended to simulate a bus traveling with an open window. In this simulation, the wind is so strong that the CO_2 can only expand directly behind the CO_2 source. Furthermore, the lower infected concentration air dispersed through the bus much faster, eventually overpowering all CO_2 emissions which can be seen in the third window of Figure 11.

Finally, the risk cell level is relatively similar across all three simulation with only slight variations due to the pathing. The number of susceptible CO_2 did not increase from the dynamic infected model. This is likely because the wind feature helped dissipate the CO_2 more efficiently. The inclusion of wind in the simulations helped reduce the number of susceptible CO_2 sources at the end of the model. This further suggests that air flow in enclosed spaces is essential to ensure proper ventilation and reduce the possibility for COVID-19 infection.

6.0 Future Work

Despite the iterative fashion of implementation, there are still many shortcomings of the model that require further improvement. Two aspects in particular affect the realism of the model: the pathing of the CO_2 sources and the rules for infection. Currently, the CO_2 source pathing functions by having a priority direction based on the position of its objective. This works great in the lab simulation yet has problems with the bus configuration. The bus configuration has workstations, represented by seats, spanning along both walls. This causes problems since the priority direction pathing sometimes sends CO_2 sources along the right wall, under a chair, before traveling up to the door, where they remain stuck because the pathing algorithm either wants to travel up or to the right, but the only option is to move to the left away from its objective. This could be fixed in future iterations by designating certain cells as path cells, where the CO_2 sources will attempt to follow the path until it needs to defer to attain its objective. This would also reduce the sporadic nature of the CO_2 sources, rendering a more realistic flow of movement.

The second improvement is to establish more concrete and realistic rules of infection. Currently, a CO_2 source must be within a threshold concentration for a set exposure time. This models the need to attain a high enough viral load to overwhelm the antibodies and render a person sick. However, the time and rate at which a person must be exposed is different for each individual, and the configuration used in the simulation doesn't represent a realistic average exposure concentration and time. This would require further analysis of the average concentration of contagion exhaled while breathing, followed by the average exposure time and concentration needed. This data is out of the scope of the project but remains a shortcoming of the model.

Finally, the model's validity is based on the expected behaviour of implementation, rather that its realism. This model requires further testing against known systems and behaviour, to validate its authenticity to a real system. The model is representative of the preliminary research and expected outcomes but is not tested against a known system. This, however, can only be undertaken once the pathing and realistic rules for infection are implemented.

7.0 Conclusion

The iterative implementation of the model allowed for the establishment of a strong foundation by individually testing each new feature to confirm proper functioning. The first iteration, dynamic susceptible model, is comprised of "CO_2 Model with Moving Occupants" and "Model of CO_2 as a Proxy for Infection", both of which are built upon the original CO_2CS model from [4]. The dynamic susceptible model models the worst-case scenario for COVID-19 infection, assuming that all CO_2 emissions carry contagion. It functioned as expected but demonstrated a serious shortcoming where CO_2 models could infect themselves. The second iteration addressed the previous iteration's shortcoming by adding two new features. The introduction of the infected CO_2 source allowed for the modelling of only the CO_2 that could contain contagion. However, using the lab configuration, the CO_2 sources were too spread out and the ventilation was too efficient for anyone to become susceptible. For this reason, a new setting of a bus was introduced. This much more confined setting with a lack of ventilation proved to be dangerous for its occupants since many CO_2 sources became susceptible. The final iteration and model, the dynamic infected complex model, added a layer of complexity and realism to the model. The third iteration now modeled both large and small droplet contagion by establishing a risk level to air cells based on their distance from the infected sources. Within a meter, a CO_2 source became 4 time more likely to become infected, and within two meters, they were twice as likely, modeling the higher quantity of contagion within large droplets and its higher risk. The second addition was wind. The introduction of an air current inhibited CO_2 containing the contagion to spread reliably in an enclosed setting, and further improved the efficiency of ventilation. This established a large emphasis on the importance of air flow within closed spaces to ensure that all of the air in a room is properly ventilated, instead of just the local air around the ventilation unit.

The final model of the dynamic infected complex model models the risk of COVID-19 infection via aerosolized contagion in an indoor dynamic setting by combining and modifying "CO_2 Model with Moving Occupants" along with "Model of CO_2 as a Proxy for Infection". The model is, however, still far from being realistic and requires the changes described in the Future Work section to produce a more applicable model and simulation.

References

[1] Ontario, P. "Updated iPAC recommendations for use of personal protective equipment for care of individuals with suspect or confirmed COVID-19. 2020."

[2] D. J. Fluxman, "'This is a no-brainer' - airborne spread of Covid-19 and the need to change official guidance," Docjon.org, 21-Jul-2020. [Online]. Available: https://www.docjon.org/post/the-evidence-for-airborne-transmission-of-covid-19-and-why-who-cdc-and-phe-advice-should-change. [Accessed: 18-Dec-2020].

[3] R, Goldstein, G. A. Wainer, A. Khan, "The DEVS Formalism", IGI Global, 2014

[4] K. Khalil, G. A. Wainer, Z. Dunnigan, "Cell-Devs Models For Co2 Sensors Locations In Closed Space", *Proceedings of the 2020 Winter Simulation Conference,* 2020 [5]

[5] https://github.com/SimulationEverywhere-Models/Cell-DEVS-CO2_spread_indoor/tree/f5847a80f04e7d7b663d16a49824fb63dd2562be