



# A Discrete-event modeling method to study human behavior for spread of diseases on university campuses

Hazel Griffith, Cristina Ruiz-Martin<sup>\*</sup>, Gabriel Wainer

Carleton University, 1125 Colonel By Dr. Ottawa, ON, K1S 5B6, Canada

## ARTICLE INFO

### Keywords:

DEVS  
Human Behavior  
Multiplex Dynamic Networks  
Agents Based Modeling  
Disease Spread

## ABSTRACT

The COVID-19 pandemic has highlighted the importance of defining sound policies to make attending workplaces safer. Sometimes, deciding on different policies is challenging as this highly depends on the behavior of the individuals. This research introduces a Discrete Event-based methodology and a prototype implementation to study such policies, including human behavior along with information about the workplace layout and building characteristics such as ventilation rate or room capacity. The method is based on a combination of agent-based models, diffusion processes and discrete-event simulation. We exemplify how to use this method using a case study based on Carleton University's Campus, in which we use the methodology and tools to study the effect of ventilation, as well as the application of a policy where sick students are denied entry to the campus on the number of disease cases on campus.

## 1. Introduction

The COVID pandemic has revealed that crowded events or locations are hot spots for the spread of airborne viruses. When functioning at normal capacity, workplaces, including university campuses, can be considered crowded places where people congregate for an extended period of time, usually for eight hours or more. For example, Carleton has over 30 000 students attending lectures and traveling the halls. Some lectures have over 300 students sitting in a closed space for at least 180 min.

On top of complying with the regulations, workplaces and university campuses cannot afford to get the disease spread among their employees or students because that will highly impact the daily activities of the organization. Therefore, a safe environment should be provided.

To provide a safer environment, Universities need to define different policies to ensure the quality of health on campus. This is also valid for most workplaces, and, as we could see during the COVID-19 pandemic, achieving these goals is complex. One way to help with the decision-making process is by using real world data on how different policies affect the spread of the disease. Nevertheless, collecting this information using real-life experiments is not viable as we may risk the health of the employees/students, seeing the effect of policies takes time, and many variables may affect the result in unpredictable ways. Instead, we may conduct a variety of experiments safely by using modeling and

simulation studies. Using these models, we can run experiments to study the effect of different parameters on the spread of the disease. For example, we could consider the ventilation rates in rooms or offices, the number of infected people attending work, or different room capacity limits. The data obtained from these experiments can be used to inform new strategies, policies, and identify hotspots.

Although there are already many models to study the spread of COVID19, there is limited work focusing on workplaces, including human behavior, workplace layout, and building characteristics in the study. For example, there are many agent-based models that take an individual's social networks into account when considering who they might transmit the disease to (Lorig et al., 2021). But they do so on the scale of a city which is great for testing government policies, but not especially helpful for a workplace that is trying to test their own disease prevention policies.

In this research, we address this limitation by building a model that includes all those aspects. More specifically, we developed a method to study the spread of diseases in workplaces considering human behavior and workplace characteristics, such as the probability of a person wearing a mask around a person who is a stranger to them, or the ventilation quality of a frequently visited room in the workplace (in the rest of the paper, we use the word stranger to refer to a person that someone does not know or is not familiar with). We base our method on the Architecture to study Diffusion Processes in Multiplex dynamic

<sup>\*</sup> Corresponding author.

E-mail addresses: [hazel.griffith@carleton.ca](mailto:hazel.griffith@carleton.ca) (H. Griffith), [cristinaruizmartin@sce.carleton.ca](mailto:cristinaruizmartin@sce.carleton.ca) (C. Ruiz-Martin).

<https://doi.org/10.1016/j.cie.2024.110732>

networks (ADPM) presented in (Ruiz-Martin et al., 2018). We adapted the architecture to study the spread of diseases in a workplace such that it could be adapted to any workplace. This was accomplished by abstracting a workplace into a series of rooms and creating generic models of the people who occupy those rooms. We then are able to instantiate those models to match the characteristics of specific rooms within a workplace, like its dimensions and ventilation rate or a specific person's behavior. We exemplify the use with the case studies of the Carleton University Campus. These case studies use information about Carleton University's campus and student schedules to conduct experiments. We show how the risk of becoming exposed to COVID-19 may be correlated with the ventilation quality of the rooms on campus and the use of the architecture to test a policy where sick students are not allowed on campus by comparing the probability of exposure to COVID-19 when the policy is and is not implemented.

The rest of the paper is organized as follows. In section 2, this paper will examine similar studies where COVID-19 transmissions were modeled. In section 3, we explain the method devised to model the spread of a disease at the workplace. In section 4, we shall focus on the model definition as an extension of the ADPM model. Section 5 introduces a discussion on model implementation. In section 6, we discuss how to use the methodology and the model focusing on the spread of COVID-19 at Carleton University. The section presents an evaluation of the effects of ventilation and a stay-home-when-sick policy on the spread of the disease. Finally, section 7 presents the conclusions and plans for future work.

## 2. Related works

In this section, we present how the spread of COVID-19 has been studied in the literature, remarking on the gaps that our research covers.

Different models have been built to study the spread of disease during the COVID-19 pandemic. Such simulations were useful for determining the impact of COVID-19 depending upon many factors, which include control policies, the area's physical layout, citizens' mobility, etc. Many of these models are based on the well-known Susceptible-Infected-Recovered (SIR) models (Ross, 1916; Ross and Hudson, 1917). These SIR-type models (where S means susceptible, I infected, and R recovered) have evolved, and now they include other states (e.g., exposed, deceased) as well as geographical level transmission dynamics (Hethcote, 2020; Barman et al., 2020; Lin et al., 2021; Sen and Sen, 2021; Davidson et al., 2023).

For example, (Hethcote, 2020) examines a mathematical model of the spread of infectious diseases, through threshold theorems for the reproduction number. It uses a basic SIR model and two advanced MSEIR and SEIR models. The M in MSEIR represents the portion of the population that is passively immune from maternal antibodies, and the E in both models (and moving forward) represents the portion of the population that has been exposed to the virus. In (Barman et al., 2020), the authors examine different strategies for limiting the spread of COVID-19 in India by using a SEIR model with a social contact matrix. The social contact matrix represents how people from different age groups interact in India specifically. The paper (Lin et al., 2021) uses SQUIARD and SIARD models to forecast the spread of COVID-19 in US, Brazil, South Korea, India, Russia, and Italy. The Q represents the quarantined population undergoing regular screening. The A represents those who are asymptomatic, and D represents the deceased. In (Sen and Sen, 2021), a SIRD model is used to analyze the time series data for five countries. Their method allows them to simultaneously analyze the temporal progression of the model for each country. Lastly, in (Davidson et al., 2023), they delve into the spatial dynamics of spreading disease by using a geography-based Cell-DEVS model. This considers how a disease spreads when a population's movement is restricted, or when a population disobeys movement restrictions.

Other studies include the use of a novel Monte Carlo simulation procedure for modeling the spread of COVID-19 over time (Xie, 2020).

These studies focus on simulating the rate at which cases will appear in a country (in (Xie, 2020), Australia and the UK) based on knowledge of the virus and the initial number of cases over a series of arbitrarily created scenarios. They calculated the day when the number of new cases per day would peak for both countries. Their results of this study were found to be accurate, indicating that their model could be applied to other nations and pandemics.

Using a finer grain, (Aguilar et al., 2007) focused on the impact of urban structure on the spread of COVID-19. Instead of focusing on how the number of cases would rise across an entire nation, they focused on individual cities and how their physical layout affects the ability of citizens to respond quickly to mobility-related policies. Such policies include stay-at-home orders, social distancing, etc. They used an SIR-type compartmental model where simulated individuals flow between compartments that may be labeled as susceptible, infectious, or recovered. They found that while densely packed cities experience a greater initial infection rate, they are also easier to enforce and transmit mobility policies to citizens. Their results proved that investing resources in early monitoring and prompt ad-hoc interventions in more vulnerable cities could make future pandemics easier to contain.

Many other works, such as (Lorig et al., 2021), consider human behavior and social relations, such as workplace, profession, or household, among others. To be specific, (Abdollahi et al., 2020) uses an agent-based model that divides the population into age groups and then considers inter and intra-group contact patterns to simulate disease transmissions. Their results show that school closures may have a limited impact on reducing the spread of COVID-19. The authors of (Akbarpour et al., 2020) developed a heterogeneous-agents network-based model where agents represent people with different social networks, occupations, and health statuses, all living in areas with different population densities, to model how effective alternative social-distancing policies are at limiting the spread of COVID-19. The authors of (Possik, 2023) created a distributed simulation system to improve their effectiveness and usability for studying disease mitigation policies. They combined an agent-based discrete event model that measures the physical contacts between patients and staff in a COVID-19 intensive care unit (ICU) with a virtual reality model that depicts a COVID-19 ICU ward. They found that combining these two models allows end-users to easily examine different disease mitigation policies for managerial operations and staff training.

There are also numerous studies on how COVID-19 is transmitted indoors. Many use the Wells-Riley model for the transmission of infectious diseases (Stephens, 2013). Research in (Aganovic et al., 2021; Li et al., 2022) both use modified versions of the Wells-Riley model. The work presented in (Aganovic et al., 2021) changed the model to estimate the impact of relative humidity on the removal of respiratory droplets containing COVID-19. Then, their experiments compared the effects of a room's ventilation quality and relative humidity on how well they reduce airborne levels of SARS-Cov-2. Their results showed that the ventilation quality of a room decreases the airborne levels of SARS-Cov-2 far more than a room's relative humidity. The work presented in (Li et al., 2022) makes a different modification to the Wells-Riley model by including how pathogens move over time through an indoor space. This is done with Computational Fluid Dynamics (CFD) simulations of airflow and aerosol transport.

They use the model to judge the effectiveness of different prevention methods on different COVID-19 variants and their results show that a combination of optimal ventilation and respiratory PPE are necessary in indoor settings. Other models forego the Wells-Riley model for a different method that entirely focuses on the transport of aerosols indoors. The work presented in (Hussein et al., 2021) develops their own indoor aerosol model combined with a regional inhaled deposited dose model to examine the indoor transport of aerosols from a person infected with COVID-19 to a susceptible person and assess the potential inhaled dose rate of particles. Their results found that with even marginally improved ventilation the dose rate declines significantly. And (Liu et al.,

2021) develops a simulation using CFD for indoor airflows and associated aerosol transport in a restaurant setting to pinpoint the environmental factors that influence the risk of infection of COVID-19. Their results show the necessity of preventive measures in indoor settings and the capability of high-fidelity CFD tools for airborne infection risk assessment.

Other methods include the use of discrete-event models using Discrete-Event systems Specification (DEVS) (Zeigler et al., 2019) and Cell-DEVS (Wainer, 2009). These methods provide some advantages such as separation of concerns between the model and the simulation, as well as ease of modeling and improved verification. For example, Cell-DEVS has been used to define an SIR model of the spatial spread of COVID-19 across South Korea (Cárdenas et al., 2021). In (Fahlman et al., 2021), the authors used an extended SIR model with Cell-DEVS to simulate the effects that an asymptomatic, yet infectious, portion of the population will have on the spread of COVID-19 at a provincial and city level. In (Fahlman and Wainer, 2024), human behaviour is integrated with a Cell-DEVS model to study its impact on the spread of COVID-19 across the provinces of Canada. They considered behaviours as mask-wearing and lockdown fatigue. Finally, in (Altamimi et al., 2021), the spread of COVID-19 is examined on a much smaller scale with a Cell-DEVS model for studying the infection risks of COVID-19 within enclosed spaces. The results were then integrated with Building Information Modelling software (Azhar et al., 2012).

As discussed earlier, we are interested in specific evaluations for studying the spread of COVID in the workplace (with a focus on University campuses). A number of articles have been published on COVID-19 mitigation strategies implemented on university campuses. Some of them (Ghaffarzadegan, 2021; Lopman et al., 2020; Bahl et al., 2021; Zhou et al., 2021; Hekmati et al., 2022) use agent-based modeling techniques, while others adapt the SEIR epidemiological model (Gressman and Peck, 2020; Borowiak et al., 2008; Ghaffarzadegan, 2021; Bahl et al., 2021; Hekmati et al., 2021). In (Zhou et al., 2021; Muller and Muller, 2021), the R reproductive number metric is used. Whereas in (D’Orazio et al., 2021), a well-mixed indoor air virus transmission model is presented. Other research has assessed this risk using high-level probabilistic models (Vecherin et al., 2022). Table 1 summarizes these articles in terms of their objectives, the simulation methodology used, and specific applications.

In this paper, we address some of the problems in previous research introducing a new method to study the spread of diseases in workplaces, considering human behavior and workplace characteristics. We use DEVS to create discrete-event models that can be used to simulate how COVID-19 (or other airborne diseases) would spread in such environment. To do so, we use the ADPM architecture, which combines a network model and an agent-based model used to study how an element may spread within a medium formalized through a transformation to DEVS. In this paper, we assume that the reader is familiar with the fundamentals of DEVS and agent-based modeling. Thus, just a brief introduction is presented. A detailed explanation of DEVS can be found in (Zeigler et al., 2019). A detailed explanation of agent-based modeling can be found in (Macal and North, 2005).

### 2.1. The DEVS formalism

A real system modeled with DEVS (Zeigler et al., 2019), is described as a composite of submodels, each of them being behavioral (atomic) or structural (coupled). Each basic model consists of a time base, inputs, states, outputs, and functions to compute the next states and outputs. As the formalism is closed under closure, coupled models can be integrated into a model hierarchy. A DEVS atomic model is formally described by:

$$M = \langle X, S, Y, \delta_{int}, \delta_{ext}, \delta_{conf}, \lambda, D \rangle$$

X:input events set;  
S:state set;

**Table 1**  
Synthesis of Research Papers on Simulating Disease Transmission.

Paper	Objectives	Simulation Methodology	Specific Applications
Simulation-based what-if analysis for controlling the spread of COVID-19 in universities (Gressman and Peck, 2020)	To develop a simulation model to analyze the spread of COVID-19 in universities	Markov chain Monte Carlo simulation method with an adaptation of the SEIR model	Applied to fictional American university with average attributes
A model of COVID-19 transmission and control on university campuses (Borowiak et al., 2008)	To provide a framework for testing different disease mitigation strategies in universities	SEIR deterministic compartmental transmission model	Applied to Emory university while using screening and testing strategies for disease mitigation
Modeling COVID-19 spread in small colleges (Ghaffarzadegan, 2021)	To study disease mitigation strategies of SARS-CoV-2 outbreaks in small residential colleges	Agents representing people follow an adaptation of the SEIR model while traversing a star graph representing spaces in a college	Applied to a small fictional college campus with various disease mitigation strategies
An agent-based model for simulating COVID-19 transmissions on university campus and its implications on mitigation interventions: a case study (Lopman et al., 2020)	To study disease mitigation strategies for university campuses	Agent-based disease transmission with a contact network and transmission mechanism	Applied to a fictional average American university campus where course modality shifts, social distancing, mask wearing, and vaccination effectiveness are tested
Simulating COVID-19 classroom transmission on a university campus (D’Orazio et al., 2021)	Study the risk of holding in-person classes by simulating airborne transmission of SARS-CoV-2 in an enclosed classroom	Well-mixed indoor air room model for airborne virus emission and exposure adapted to a classroom	Applied to the University of Southern California with various disease mitigation strategies
Simulating COVID-19 in a university environment (Bahl et al., 2021)	To study the effectiveness of disease mitigation strategies given a COVID-19 outbreak on a university campus	Full-scale stochastic agent-based model that uses an adaptation of the SIR model	Applied to a fictional university with various disease mitigation strategies
Controlling the spread of COVID-19 on college campuses (Zhou et al., 2021)	To determine how students can return to classes safely amidst the COVID-19 pandemic	Uses a household model and a stochastic agent-based model that uses the R reproductive number metric	Applies the household model to Duke University’s dorms to test the effectiveness of giving every student a single room And the stochastic agent-based model is used to find the class size threshold beyond which classes should be moved online
Simulation-Based Analysis of	To study the airborne COVID-	Uses the R reproductive	Applies the model to a

(continued on next page)

Table 1 (continued)

Paper	Objectives	Simulation Methodology	Specific Applications
COVID-19 Spread Through Classroom Transmission on a University Campus (Muller and Muller, 2021)	19 transmission risk of holding in-person classes at a university campus given different disease mitigation policies	number metric with an airborne transmission risk model in an enclosed room	dataset obtained from a large US university and tests various disease mitigation strategies
Mathematical modelling of the spread of COVID-19 on a university campus (Hekmati et al., 2021)	To study various disease mitigation strategies for preventing an outbreak on a college campus	Adapts a SEIR model for a deterministic transmission dynamic compartmental	Applies the model to the estimated data from the Villanova University COVID Dashboard for the 2020 fall semester and tests various disease mitigation strategies
A probabilistic model to evaluate the effectiveness of main solutions to COVID-19 spreading in university buildings according to proximity and time-based consolidated criteria (Hekmati et al., 2022)	To study various disease mitigation strategies for COVID-19 for university campuses	Uses agent-based modelling techniques with consolidated proximity and exposure-time-based rules	Applies the model to a case study for the faculty of engineering campus at università Politecnica delle Marche of Ancona Italy
An Agent-Based Modeling and Virtual Reality Application Using Distributed Simulation: Case of a COVID-19 Intensive Care Unit	To improve the effectiveness and usability of COVID-19 transmission simulations by developing a distributed simulation system that can combine different COVID-19 simulations	The distributed simulation system combines an agent-based discrete-event contact tracing model with a virtual reality model	The model is used to depict physical contacts between patients and staff in an intensive care unit with the purpose of studying disease mitigation policies

Y: output events set;  
 $\delta_{int}: S \rightarrow S$ , internal transition function;  
 $\delta_{ext}: Q \times X \rightarrow S$ , external transition function;  
 where  $Q = \{(s, e) / s \in S, \text{ and } e \in [0, D(s)]\}$ ;  
 $\delta_{con}: Q \times X \rightarrow S$ , confluence transition function;  
 where  $Q = \{(s, e) / s \in S, \text{ and } e = D(s)\}$ ;  
 $\lambda: S \rightarrow Y$ , output function; and  
 $D: S \rightarrow R_0^+ \cup \infty$ , elapsed time function.

Each model is seen as having input and output ports to communicate with other models. The input and output events will determine the values to appear in those ports. The input external events are received in an input port, and the model specification should define the behavior under such inputs. The internal events produce state changes, whose results are spread through the output ports. The ports influences will determine if these values should be sent to other models.

A basic model can be integrated with other DEVS basic models to build a structural model. These models are called coupled, and are formally defined as:

$$CM = \langle X, Y, D, \{M_i\}, \{I_i\}, \{Z_{ij}\} \rangle$$

$X_i$  is the set of input events;  
 $Y_i$  is the set of output events;  
 $D_i$  is an index for the components of the coupled model, and  $\forall i \in D$ ,  $M_i$  is a basic DEVS model, where  $M_i = \langle X_i, S_i, Y_i, \delta_{int_i}, \delta_{ext_i}, \delta_{con_i}, \tau_{a_i} \rangle$  or CM.  
 $I_i$  is the set of influences of model  $I$ , and  $\forall j \in I_i$ , and  
 $Z_{ij}: Y_j \rightarrow X_i$  is the  $j$  to  $i$  transition function.

### 3. A method to model the spread of COVID-19 at the workplace

As discussed earlier, our objective is to study new methods based on discrete event modeling simulation to model how COVID-19 spreads in workplace environments (with a focus on University Campuses). The proposed method is based on an Architecture to study Diffusion Processes in Multiplex dynamic networks (ADPM), introduced in (Ruiz-Martin et al., 2018). ADPM was designed to study diffusion processes, which are used to model how an element may spread within a given medium. For example, one could consider the diffusion of traffic across a network of roads, or how gossip travels between people in social media. The medium through which the element travels and how it travels is determined by a multiplex dynamic network (Gómez et al., 2013). Multiplex dynamic networks are those in which the connections among nodes change over time, and different connections have different properties. In ADPM, these connections are divided in two types: direct links and indirect links. The first one represents the direct transfer of the element from one node to the other (agent to agent), and for indirect links a device or medium is used to establish the link between agents. For instance, a direct link could represent two people talking face to face whereas an indirect link can represent two people talking over the phone. The multiplex part of the network is modeled by using different devices or mediums.

Fig. 1 shows the workflow to create a model and a simulator using ADPM. Starting with data collection for a diffusion experiment, it leads into creating a document including model requirements and assumptions. This document is further used to create a network model (specifying the structure of the system of interest) and an agent-based model (that defines the behavior of different entities). Once model cross-verification is completed, they are used in tandem to create a diffusion abstract model in which agents connected in networks spread an element. This model is then computerized to create a simulator that produces logs for the different simulation scenarios under study. This entire process can then feedback to the start where analysis reports of the simulation logs can be used to validate the results with domain experts and technical staff. This process can loop as many times as necessary to validate the results.

When we combine the network model and the agent-based model to get the diffusion abstract model, a model with the structure presented in Fig. 2 is built (this process is automated and takes as input the network model and the ABM). Each part of the architecture presented in Fig. 2 has a unique purpose. The Node model represents the agents that transfer an element, like people spreading gossip or computers sharing a virus. The Node Updater may be used to provide dynamic information to the nodes, like a change in weather or power outages. The Diffusion Element Generator creates instances of the element to be spread between nodes, like the virus in a computer.

The Direct Link model represents direct interactions between agents where nothing was used to facilitate the interaction (i.e., two people talking face-to-face). The Direct Link Updater is used to change parts of the direct links at runtime (for instance, environmental noise when talking in person). The Indirect Link model represents connections between agents that are facilitated with a device, like a phone call. The Indirect Link Updater then provides changes to the indirect link communications, like changing the rules of a social media network. Link Connectors are used to provide a higher level of granularity in the

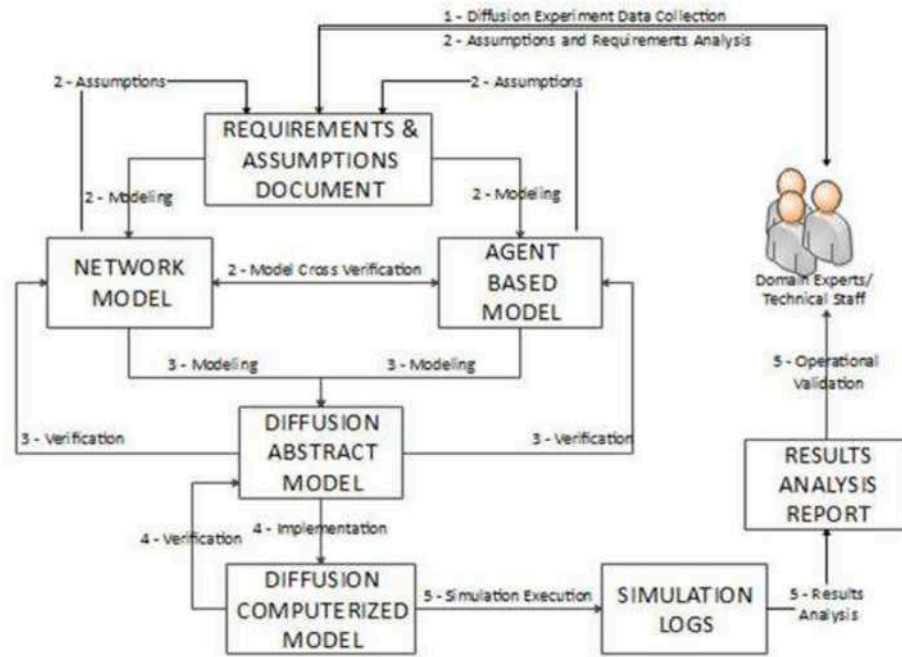


Fig. 1. ADPM Workflow (Ruiz-Martin et al., 2018).

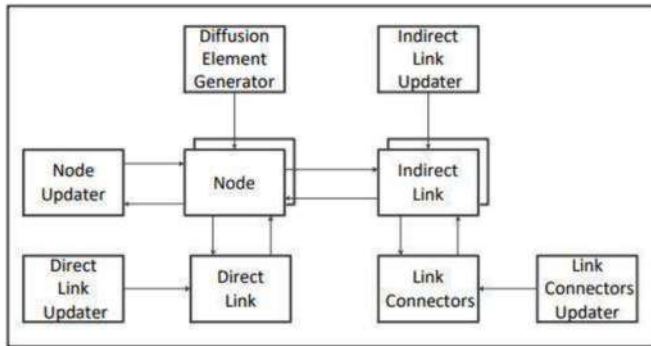


Fig. 2. The ADPM model architecture (Ruiz-Martin et al., 2018).

model. They specify how the different devices are connected to each other. For example, phone calls can happen through voice-over IP or a cellular network. Lastly, the Link Connectors Updater provides runtime changes to the link connectors, such as an internet outage.

The workflow in Fig. 1 is directly applicable to the problem of COVID-19 (or other airborne disease) in workplaces. We study this problem as a diffusion process where the element being diffused is an airborne disease, and the agents are the people infected (using different links for friends, co-workers, strangers, etc.). We adapt ADPM (which is generic) to satisfy the problem’s specific characteristics. We decide what parts of the ADPM are relevant to the problem and we define those components for our application, which will be detailed in section 4. The architecture adapted from the ADPM model architecture can be seen in Fig. 3.

The Node component represents a Person (i.e., individual’s behaviors) as all nodes in our model are people. The NodeUpdater is used to modify how people’s behavior changes due to external circumstances. In our case, we are only interested in modeling the change of behavior due to changes in the weather. Therefore, we label Node Updater as Weather. If we were to include other external factors that would modify the person’s behavior (e.g., news on the TV or social media) we would have to update and rename this component. The Direct Link component is mapped to a Room component that represents the direct connections

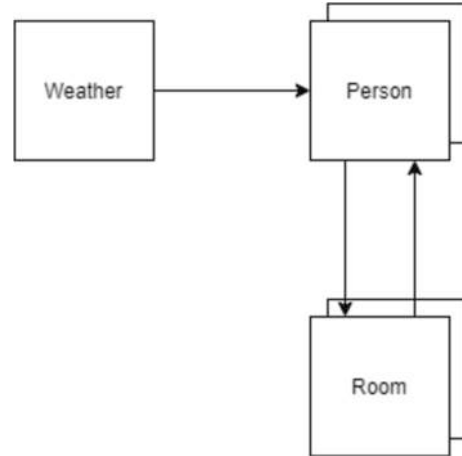


Fig. 3. ADPM model architecture adapted for COVID spread models in workplaces.

between the nodes (i.e., people). The Direct Link Updater is not needed as the Room component will not change due to external factors (e.g., the size of the room will not change). In our model, all the components related to indirect links are not needed. In the case of a virus being spread across a population, a direct link could represent sneezing near another person, while an indirect link could represent an infected person handling an item and then that item being handled by another person. For the spread of COVID, we will not be considering indirect links because it has been found that the odds of a person becoming sick through indirect contact with a sick person is very low (Zhang et al., 2020).

#### 4. Applying ADPM in the workplace

This section focuses on the ADPM model specification and its definition for modeling the diffusion behavior of COVID-19 (or another airborne transmissible disease) in the workplace. We model a number of people moving around several rooms (or going home) during a certain

period of time. Each person will have a general relationship established with each other person to realistically model how they choose whether to engage in COVID-safe behavior. The behavior under examination can include whether people would wear masks, socially distance, and obey maximum occupancy policies in different scenarios.

The rooms will accurately resemble the workplace being studied, and each person's schedule will accurately resemble the schedules of people with different workplace roles. It will consider general weather patterns, such as rain and snow, that influence each person's travel decisions. For example, if it is raining, it is more likely a person will travel indoors to reach their destination instead of going outside.

The model includes the CO2 concentration in rooms, the probability of each person being exposed to COVID-19 once they leave a room, each person's location at any given timestamp, and the travel routes that each person chooses when moving between rooms.

#### 4.1. Person Agent-Based and network models

We define a variety of human behaviors exhibited within the model, including wearing masks, social distancing, entering a room at maximum occupancy, or traveling on a route that is more or less likely to expose them to the virus. The first three behaviors are determined by each Person's relationship status with every other Person in the model (as individuals will act differently around a close friend than around a stranger). The Person's agent-based behavior is defined using an XML file shown in Fig. 4. This model also includes the person's connections.

*DecisionMakerBehavior* is an object with many fields. *ID* uniquely identifies a person, after which we define the person's *location*. The

*currStartTime* and *timeInFirstLocation* are used to define how much time the person has left in their first location (time 0 which represents midnight and we add them modulus of 1440; the number of minutes in a day. For example, in Fig. 4, *currStartTime* is 1280 and *timeInFirstLocation* is 760, therefore their sum mod 1440 equals 600. This means there are 600 min until their next scheduled event). Then, *isSick* defines if the individual is sick; *exposed* is used to define if they have been exposed to COVID-19 and then if they have been *vaccinated*. The *wearingMaskCorrectly* is used to define if they are wearing a mask over their mouth and nose, and *socialDistance* is used to define if they practice social distancing. The *weatherThreshold* indicates how bad the weather must be for this person to avoid travelling outdoors. This could take many forms depending on the workplace being examined. For example, if a large workplace has open air walkways between locations, then poor weather might incline a Person towards taking a different route indoors where they are more likely to be exposed to the virus. The *relationship* object stores a list of the *IDs* of the other people in the model and this person's relationship with them (friends, acquaintances, or strangers). These categories determine how likely the person is to adhere to safety policies around them. The *behaviourRulesPerson* object stores the probabilities of a person performing each action around people of each relationship category. Lastly, the *locationPlan* object stores the name of each room the person will visit in their daily schedule, what time they will enter each room, and how long they will be in each room. The *home* tag is used to indicate they have left campus. *Outdoors* defines whether people travel along a safer path to reach their destination or take an enclosed route that may expose them to the virus.

While most of the attributes are static (i.e., instantiated when the

```
<DecisionMakerBehaviour>
  <ID>3</ID>
  <location>home</location>
  <currStartTime>1280</currStartTime>
  <timeInFirstLocation>760</timeInFirstLocation>
  <isSick>False</isSick>
  <exposed>False</exposed>
  <vaccinated>False</vaccinated>
  <wearingMaskCorrectly>True</wearingMaskCorrectly>
  <socialDistance>True</socialDistance>
  <weatherThreshold>6</weatherThreshold>
  <relationship>
    <relationship ID="1" type="friends" />
    <relationship ID="2" type="friends" />
    <relationship ID="4" type="acquaintances" />
    <relationship ID="5" type="acquaintances" />
  </relationship>
  <behaviourRulesPerson>
    <personRelations status="acquaintance" safeDistanceProb="50" maskWearingProb="50"
      enterMaxOccRoomProb="50" />
    <personRelations status="friends" safeDistanceProb="20" maskWearingProb="20"
      enterMaxOccRoomProb="80" />
    <personRelations status="strangers" safeDistanceProb="80" maskWearingProb="80"
      enterMaxOccRoomProb="20" />
  </behaviourRulesPerson>
  <locationPlan>
    <locationPlan room="38-VS - 2285" timeinroom="90" startTime="610" />
    <locationPlan room="Outdoors" timeinroom="10" startTime="700" />
    <locationPlan room="home" timeinroom="490" startTime="710" />
    ...
  </locationPlan>
</DecisionMakerBehaviour>
```

Fig. 4. ABM model of a person stored in an XML file (Griffith, 2022a).

model is defined), the variables *sick*, *exposed*, *location*, and *currStartTime* will evolve over time based on the person’s behavior. The XML file is used to initialize those variables.

#### 4.2. Agent-Based model of a room

Each room is also defined using an XML file that contains seven unique characteristics. Most of this data can be gathered from BIM models of the rooms where available, or through floor plans. If some of this information is not available, they can be defined as parameters of the model.

Fig. 5 shows a Room’s XML file. First, we have an *ID* that indicates a unique name of the room. A *ventilationRating* represents the air changes per hour (ACH), and the *socialDistanceThreshold* includes the maximum number of people that can be in a room and still maintain social distancing. Then, *maxOccupancy* is the maximum number of people that can be in the room according to policy. The *wearsMaskFactor*, *socialDistanceFactor*, *vaccinatedFactor*, and *sickPeopleCO2Factor* affect how likely it is a person will become exposed to COVID-19. The *highCO2FactorThresholds* are the levels of CO2 concentration in a room that indicate when a person’s exposure to COVID-19 is becoming more likely. The *highCO2Factors* are used to influence the probability of exposure if its associated *highCO2FactorThreshold* is reached by the CO2 concentration within the room. The *respIncreasePerMin* is a calculated average amount of CO2 a person breathes out every minute. Lastly, the *squareMetres* and *height* fields indicate the dimensions of the room used for calculating the CO2 concentration.

#### 5. DEVS Representation: ADPM definition for COVID spread

Fig. 6 shows the structure of a DEVS coupled model derived from ADPM presented in Fig. 3. The model consists of three main components: (1) the Weather atomic model, (2) the Person Coupled Model, and (3) the Room and Outdoors Coupled Models. Note that the Outdoors coupled model is just an instance of a room model where the probability of getting COVID-19 is neglectable.

The **Weather atomic model** generates values that represent the quality of the weather every 24 h of simulation time. In our case, the model is connected to every instance of the Person Coupled model. The **Person Coupled model** is composed of two atomic models: **Person** and **FilterProbGetSick**. The number of instances of the Person Coupled Model is the number of people in the model. The Person atomic model represents each person node within the model. It models their behavior with daily schedules and interpersonal relationships using the ABM model attributes defined in the XML in section 4.1. It also tracks their chances of being infected or sick, and how that affects their routine. It

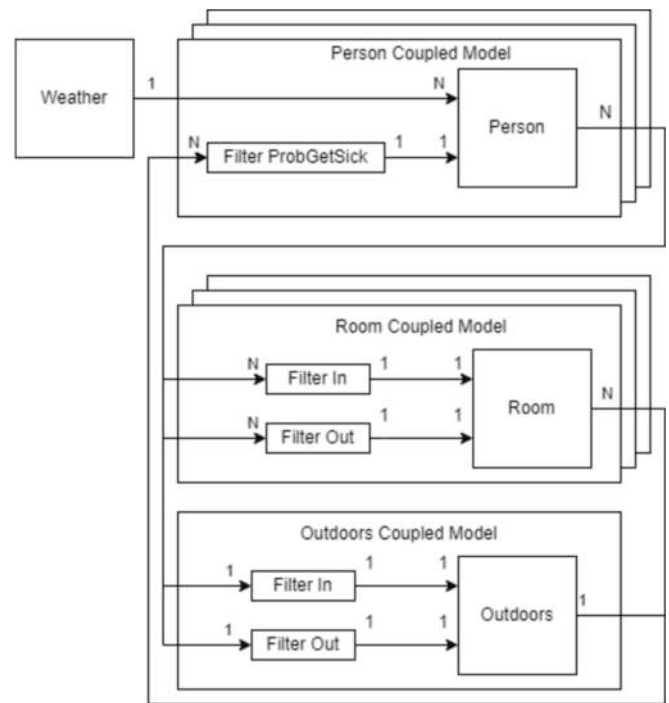


Fig. 6. DEVS Top Model.

receives inputs with the probability of exposure to the virus and the current weather conditions. The model transmits personal information (Id, relationship types, probabilities of acting with different behaviours, and information about sick, vaccinated, practicing social distancing, and wearing a mask states). The model follows the schedule loaded from the XML and determines the room the person needs to enter. The travel path (i.e., the safer or riskier route), is chosen by checking if the current weather value is less than the person’s weather threshold.

Once the scheduled time spent traveling has concluded, an internal transition determines what room the person has arrived. The model will then receive a message through its input port (the one connected to the FilterProbGetSick model) containing the probability of the person being exposed to the virus while traveling. When the scheduled time in the room is over, another internal transition occurs. If the person is considered exposed, a random value is generated, and if it is greater than a threshold representing the probability of becoming sick after exposure, the person will become sick. Another randomly generated value is used to determine how many days before the person is considered sick; this

```
<RoomParameters>
  <ID>5,RH - 1100</ID>
  <ventilationRating>3</ventilationRating>
  <socialDistanceThreshold>35</socialDistanceThreshold>
  <maxOccupancy>26</maxOccupancy>
  <wearsMaskFactor>1</wearsMaskFactor>
  <socialDistanceFactor>1</socialDistanceFactor>
  <vaccinatedFactor>-2</vaccinatedFactor>
  <sickPeopleCO2Factor>2</sickPeopleCO2Factor>
  <highCO2FactorThresholds>1001,1840,1857,1904</highCO2FactorThresholds>
  <highCO2Factors>1,2,3,4</highCO2Factors>
  <respIncreasePerMin>340000</respIncreasePerMin>
  <squareMetres>71.019997</squareMetres>
  <height>2.438000</height>
</RoomParameters>
```

Fig. 5. Room XML file (Griffith, 2022b).

represents the incubation period of the disease being studied. This process is repeated every time the scheduled time in a room or traveling between rooms is over. Once a person is exposed, at each internal transition, it checks if enough time has passed for the incubation period to end. Once this happens the person will be considered sick. Once the person becomes sick, the person will abandon their schedule at random and go to “home”. This is to represent people, who, for different reasons (like being asymptomatic) choose to go to their workplace.

Every time the weather changes, the model receives a message through its input port (the one connected to the weather atomic model). Through an external transition, the person model updates the Person model’s current weather threshold value.

Fig. 6 shows the **Room Coupled model**, which is composed of three atomic models: **Room**, **FilterIn**, and **FilterOut**. The number of instances of the Room Coupled Model is the number of rooms in the model. **Room** represents each room component within the model. This includes the enclosed riskier path a person might travel and the room destination of the person. It models their characteristics, such as CO2 concentration, and calculates the probability of exposure for people spending time in the room using the ABM model attributes defined in the XML in section 4.2. The model also records which people are within the room. Room receives messages that contain the travel and personal information of people within the model. Each room model is initialized with the ABM model attributes stored in its unique XML. At first, each room is idle and waiting for a message to arrive at an input port. An external event indicates a person is entering the room, and the external transition is

$$respIncPerMin\left(\frac{mg}{min}\right) = \left(500\left(\frac{mL}{Breath}\right) \cdot \left(\frac{35000ppm + 50000ppm}{2}\right) \cdot \left(\frac{1mg/L}{1ppm}\right)\right) \cdot \left(\frac{12\left(\frac{breaths}{min}\right) + 20\left(\frac{breaths}{min}\right)}{2}\right) = 340000\frac{mg}{min} \quad (1)$$

triggered. At the start of every external transition the elapsed time is used to calculate the current CO2 concentration of the room. Then, the person’s id, along with the rest of their information, is stored in a list that includes everyone who is currently in the room. The model uses their information to determine and store the person’s relationship status with everyone else who is in the room. If there are many messages from people indicating they are entering the room, then there is a chance that the room will be at or above its maximum occupancy. The maximum

$$CO2Concentration\left(\frac{mg}{L}\right) = CO2ConcentrationFlow\left(\frac{mg}{L}\right) + PreviousCO2Concentration\left(\frac{mg}{L}\right) \quad (3)$$

occupancy of a room is a parameter that is adjusted beforehand. If a message is received that indicates a person is entering the room, but the room is at maximum occupancy, then the room model uses information about the person entering the room to determine if they enter the room.

When an external event indicates that a person is leaving the room, the probability of that person having been exposed to the virus is calculated through an external transition. The person is matched to a rank based on their behavior. Each rank is associated with a range of probabilities of infection, and it is calculated based on CO2 concentration in the room and behavioral factors (mask wearing, social distancing, vaccination status). We are using CO2 concentration as proxy for the virus’s airborne particles (Peng and Jimenez, 2021), so the greater the concentration of CO2 in a room, the greater the risk of becoming exposed to the virus. Then, a probability of exposure is

generated within the bounds specified by the rank. The record of people currently in the room and their relationships with each other is updated to account for the persons who left. If there are people waiting outside, then the first person is added. Internal transitions clear the list of people leaving the room next.

The **FilterIn** and **FilterOut** atomic models are similar to the **Filter-ProbGetSick** model, and they only forward the message to/from the Room if the id of the room matches.

Every instance of the Room atomic model calculates the new concentration of CO2 in the room whenever a person enters or leaves the room. This calculation is completed with the formulas we discuss next. First of all, we use two constants: (1) average CO2 concentration and (2) average amount of CO2 respired by a person (*respIncPerMin*). We use an average CO2 concentration outdoors of 400 ppm, as according to Environmental Health Perspectives, “although typical outdoor CO2 concentrations are approximately 380 ppm, outdoor levels in urban areas as high as 500 ppm have been reported” (Satish et al., 2012). The average amount of CO2 respired by a person is 340,000 mg CO2 per minute. This is calculated as per (1) by multiplying the tidal volume of a breath 500 mL per breath (Hallet et al., 2021) by the average CO2 concentration of a breath: “An average adult exhale contains 35,000 to 50,000 parts per million (ppm) of CO2 on each breath” (Sidhardhan and Das, 2021) and multiplying that result by the average number of breaths per minute while at rest, “They differ from those found in periodic breathing, which have a period length of 12–20 breaths” (Pribean, 1963).

The air changes per hour (ACH) can be anywhere between 1 and 8 depending upon the room. For example, in the room XML example in section 4, the ACH is equal to 3. Next the ventilation rate per minute is calculated using (2).

$$ventilationRatePerMin\left(\frac{L}{min}\right) = \frac{ACH * volumeOfRoom(m^3) * 1000L}{60minutes * m^3} \quad (2)$$

Then the CO2 concentration is calculated using (3).

Here, *CO2ConcentrationFlow* represents the change in the CO2 concentration since the previous CO2 concentration was calculated. To calculate the *CO2ConcentrationFlow* several equations are needed. The equations (4), (5), and (6) are the CO2 entering the room from its ventilation per minute, the CO2 being breathed out by everyone in the room per minute, and the CO2 leaving the room per minute.

$$CO2InPerMin\left(\frac{mg}{min}\right) = ventilationRatePerMin\left(\frac{L}{min}\right) * averageCO2\left(\frac{mg}{L}\right) \quad (4)$$

$$CO2PeoplePerMin\left(\frac{mg}{min}\right) = respIncPerMin\left(\frac{mg}{min}\right) * numberOfPeople \quad (5)$$



$$CO2OutPerMin\left(\frac{mg}{min}\right) = ventilationRatePerMin\left(\frac{L}{min}\right) * PreviousCO2Concentration\left(\frac{mg}{L}\right) \quad (6)$$

The  $CO2ConcentrationFlow$  then uses all three of these formulas, the time elapsed, and the volume of the room, to determine the change in the CO2 concentration in that time.

$$CO2ConcentrationFlow\left(\frac{mg}{L}\right) = \frac{(CO2InPerMin\left(\frac{mg}{min}\right) + CO2PeoplePerMin\left(\frac{mg}{min}\right) - CO2OutPerMin\left(\frac{mg}{min}\right) * ElapsedTime(min))}{volumeOfRoom(m^3) * 1000\left(\frac{L}{m^3}\right)} \quad (7)$$

Using (7), the longer the elapsed time variable is, the less accurate it becomes. This is overcome by only calculating the new CO2 concentration at intervals of 10 min or less.

The CO2 concentration equations were validated in the following way. We obtained historical data with which we can validate the current model. This information consists of the CO2 concentration values at half-hour intervals over 24 h, starting from 0:23 to 23:53 of a room on Carleton University campus. The room has a volume of 439.96875 m cubed, at most 39 people were in the room, and the ACH value is unknown. The model calculates the concentration of CO2 in a room given an amount of people, and the number of people in the room at each half-hour interval is unknown. Therefore, we must first determine what the most likely distribution of people in the room across all the time slots could be. Then we must find the ACH value through trial and error. Once we achieve CO2 concentration values for each timestamp that are as close to the obtained data as possible by changing the people distribution, ACH value, and average CO2 value, we can conduct a  $t$ -test to judge if there are any inconsistencies between the historical data set and the data set the model produced.

We obtain the distribution of people across all the 48 time slots by mapping the maximum and minimum values of the CO2 concentration and amount of people onto a linear function. It would take the form of (8).

$MaxCO2Conc = 980$ ;  $MinCO2Conc = 572$ ;  $MaxNumbPeople = 39$ ;  $MinNumbPeople = 0$ .

**Table 2**

Distribution of people in the room per time slot.

Time Stamps (2019-09-10)	Number of People	Time Stamps (2019-09-10)	Number of People	Time Stamps (2019-09-10)	Number of People
0:23	0	8:23	6	16:23	5
0:53	0	8:53	11	16:53	34
1:23	0	9:23	15	17:23	32
1:53	0	9:53	15	17:53	19
2:23	1	10:23	23	18:23	4
2:53	1	10:53	34	18:53	4
3:23	1	11:23	29	19:23	4
3:53	1	11:53	4	19:53	3
4:23	1	12:23	6	20:23	3
4:53	1	12:53	9	20:53	2
5:23	1	13:23	21	21:23	2
5:53	2	13:53	39	21:53	3
6:23	2	14:23	33	22:23	3
6:53	2	14:53	25	22:53	2
7:23	2	15:23	23	23:23	2
7:53	2	15:53	18	23:53	3

$$a = \left( \frac{MaxNumbPeople - MinNumbPeople}{MaxCO2Conc - MinCO2Conc} \right) a = 39408 \quad (8)$$

$$NumberPeople = a * (CO2Concentration - MinCO2Conc)$$

Using the CO2 concentration values recorded in the room at each time slot, we calculate a distribution of people at each timeslot that can be seen in [Table 2](#).

The distribution has small numbers of people inside the room after hours. This could be represented by people cleaning or working late. We then conducted 12 tests where in each we changed the ACH value, people distribution, and average CO2 value. The resulting CO2 concentration values calculated by the test, the differences between the historical data and the data the model produced, and the squared differences between the regular difference and the difference average. This data is then used to calculate the average difference and the standard deviation of the differences.

On each test we conducted a  $t$ -test to judge whether there are any inconsistencies with the historical data and the data the model produced. We treated the CO2 concentration values as independent from one another, because the model calculates the CO2 concentration multiple times to get the actual value, it would be the same regardless of the previous CO2 concentration value. We also treated the differences as having a normal distribution, which for the purposes of the  $t$ -test is sufficient. A significance value of 0.05 was also used. Each test was done to see if the null hypothesis where the mean of the differences  $\mu$  is equal to 0 can be rejected. This is done by calculating the  $t$  statistic using (9).

$$t_0 = \frac{\bar{d} - \mu_d}{S_d/K} \quad (9)$$

As there are 48 data points a  $K$  value of 48 is used. Next, because a significance value of 0.05 is used, and  $K-1$  equals 47, a critical value of 2.01 is used. If the  $t$  statistic is less than the critical value, then no inconsistencies between the historical data and the data produced by the model could be found as the null hypothesis cannot be rejected.

Out of all twelve of the different tests conducted, only 3 of them passed the  $t$ -test and did not show any inconsistencies. The first of these three tests was test 3 with an ACH value of 4.5, average CO2 value of 575, and a people distribution that set positions 6–11 as 0, and the last four positions as 0. It passed with a  $t$  statistic of 1.54. The second test was test 4 with an ACH value of 4.5, average CO2 value of 575, and a people distribution that set positions 6–11 as 0, and the last four positions matched the mapping such that it was 3, 2, 2, 3. It passed with a  $t$  statistic of 0.662. The third test to pass was test 12, with an ACH value of 4.475, average CO2 value of 575, and a people distribution that set positions 6–11 as 0, and the last four positions were 3, 2, 2, 3. It passed with a  $t$  statistic of 1.45. The people distributions positions of 6–11 and

```

Person (const char * personFileName) noexcept{
    person.load(personFileName);
    //Initialize all state attributes of the person atomic model
    timeR = 0;
    currentWeather = 0;
    ...
}

```

Fig. 7. Constructor of the Person class.

the last four were changed in different ways, because it seems unlikely that a person would be in the room from 2:30 to 5:30, and after 10:30. The passed t-tests show that given the unknown variables are correct, and that the average CO2 value is correct, we can calculate CO2 concentration values that are not inconsistent with actual values.

The **Outdoors Coupled model** includes three atomic models: **Outdoors**, **FilterIn**, and **FilterOut**. It represents a space for people to travel through where they have a very low probability of becoming infected. It is similar to the room model, the only different is that the probability of exposure to the virus is always very close to zero.

## 6. ADPM for COVID spread in the DEVS Cadmium tool

The model in section 4 was defined as a parameterized DEVS model in C++ using the Cadmium library. The model is instantiated through XML files. The atomic models were written as C++ classes that follow Cadmium's specifications (Ruiz-Martin and Wainer, 2020). They communicate by passing three custom data structures:

- **ProbGetSick** contains a person's probability of being exposed to the disease.
- **WeatherInfo** represents the severity of the weather on a scale from 0 (bad weather) to 10 (ideal weather).
- **PersonInfo** contains information of the person traveling from one room to another: (1) the ID of a Person, (2) if the person is sick, has been exposed to the virus, has a tendency to wear a mask, has a tendency to practice social distancing, (3) ID of the room being entered, (4) the time that a person enters the new room, (5) the ID of the room the person is leaving, (6) the time that a person will leave the room they are entering, (7) a person's tolerance for inclement weather, (8) how many minutes until the person leaves the room they are entering, (9) *Relationship* objects that contain the person's

relationship type with every other person, and (10) a map that contains a relationship type, and their *BehaviourRulesPerson* objects (a person's probability of performing a behaviour given they are around only people they have one relationship type with).

All atomic models are implemented as C++ classes in the Cadmium DEVS simulator. The implementation is a direct translation of the models explained in section 4 into C++ code following the Cadmium API. The whole implementation is available in our GitHub repository (Griffith and Baggs, 2022). We will use the Person model as an example, as the remaining models presented in Section 4 are implemented similarly to this one.

We must first define the model's constructor function (Fig. 7) and how it is used to create instances of the Person atomic model. The Person constructor is used to create instances of the person atomic model. It loads the person data from the Person XML files as *DecisionMakerBehaviour* objects (person.load(personFileName)). The *DecisionMakerBehaviour* class keeps track of a Person instance's schedule, current location, and the traits that determine how likely it is that they become exposed to the virus. The class *setNextLocation* method is used to acquire the next location in the Person's schedule by using the start time of their current activity and the time they spend doing that activity. It also includes methods for loading the object from a Person XML file and another for saving the object to an XML file. Within the Person class constructor, all state variables of the atomic model are initialized using initial values or the information from the XML file.

We then define the external transition function, as seen in Fig. 8. This is executed whenever an input is received along either or both of the Person class's input ports. The function receives the messages from the input ports as well as a variable  $e$  that is the amount of time in minutes that has elapsed since the last state transition. The function processes all the messages received from both input ports. We first check *ProbGetSick*

```

void external_transition(TIME e, make_message_bags<input_ports>::type mbs){
    vector<ProbGetSick> msgBagSick = get_messages<Person_ports::infectionProb>(mbs);
    vector<WeatherInfo> msgBagWeather = get_messages<Person_ports::weatherUpdates>(mbs);
    for (int i=0 ; i < get_messages<Person_ports::infectionProb>(mbs).size(); i++){
        int r = rand()%100; //when ProbGetSick received, random number generated
        if (r <= (msgBagSick[i].probSick)){ //if r less than the probability of sick
            if (!person.isSick){ //then, then the person is now sick
                person.exposed = true; travelInfo.exposed = true;
            }
        }
    }
    for (int i=0; i < get_messages< Person_ports::weatherUpdates>(mbs).size(); i++) {
        currentWeather = msgBagWeather[i].newState;
        if (firstWeatherUpdate == true)
            firstWeatherUpdate = false;
        else
            timeR=(person.nextLocation.timeInRoomMin + person.currStartTime)%1440;
        if (remainingTimeUntilNextInternalTransition > 0)
            timeR = remainingTimeUntilNextInternalTransition;
    }
}

```

Fig. 8. External transition function of the Person class.

```

make_message_bags<output_ports>::type output() const {
    make_message_bags<output_ports>::type bags;
    get_messages<Person_ports::nextDestination>(bags).push_back(travelInfo);
    return bags;
}

```

Fig. 9. Output function of the Person class.

```

void internal_transition(){
//sets up the person to output the correct PersonInfo object at the next time interval
if (person.exposed){
    uniform_int_distribution<int> proboSickDistribution(0,100),
    int r = proboSickDistribution(generator);
    if (r > 70){
        uniform_int_distribution<int> timeUntilSickDistribution(0, 10080);
        timeUntilSick = timeUntilSickDistribution(generator);
        becomingSick = true;
    }
    person.exposed = false;
}
if (stayingHome == false){
    if ((person.isSick)&&(severityDecided == false)){
        uniform_int_distribution<int> proboSickDistribution(0,100);
        int r = proboSickDistribution(generator);
        if (r < 0) goingHome = true;
        severityDecided = true;
    }
}
if (firstTravel){ ...
// we use person.setNextLocation() with the current start and remaining times to find the
// next location. The first remaining time is loaded from the person xml file and
// initialized
}
timeUntilNextWeatherUpdate -= timeR;
if (timeUntilNextWeatherUpdate <= 0) {
    remainingTimeUntilNextInternalTransition= -1 * timeUntilNextWeatherUpdate;
    timeUntilNextWeatherUpdate = 1440 - remainingTimeUntilNextInternalTransition;
}
else remainingTimeUntilNextInternalTransition = 0;

if (person.nextLocation.roomID == "Outdoors" || person.nextLocation.roomID == "Tunnels") {
    if (currentWeather >= travelInfo.weatherThreshold)
        person.nextLocation.roomID = "Outdoors";
    else
        person.nextLocation.roomID = "Tunnels";
}
travelInfo.roomIDEntering = person.nextLocation.roomID;
person.location = person.nextLocation.roomID;
if ((travelInfo.roomIDLeaving.compare("home") == 0)&&(goingHome))
    stayingHome = true;
}
}

```

Fig. 10. Internal transition function of the Person class.

objects from the *infectionProb*. Within this section a random integer from 0 to 100 is generated, and if it is less than the probability of becoming exposed received from the *ProbGetSick* object then if the Person is not already sick, they are considered exposed to the virus. Random values are used here as calculating the exact chance of becoming exposed to the virus is outside the scope of this research. We then process *WeatherInfo* objects from the *weatherUpdates* input port. First *currentWeather* is updated with the new value, and if this is not the first update then the time remaining is set to the time until the first location change after midnight at time 0. If it was determined in the last internal transition that a message from the *weatherUpdates* input port would be received before the next internal transition occurs, then the state's *timeR* variable

is set to equal to *remainingTimeUntilNextInternalTransition*.

The output function sends the *PersonInfo* object known as *travelInfo* to the output port *nextDestination*, as seen in Fig. 9.

After executing the output function, the internal transition function, seen in Fig. 10, is triggered to reflect that the Person instance has finished spending time in a Room and are heading to their next location.

First, if the person has been exposed to the virus, a random number between zero and one hundred is generated. If the value is greater than a threshold that represents the probability of becoming sick, then they are considered to be sick. A random number between zero and the maximum incubation time for the virus is generated and stored as the time until the person will start showing symptoms. If the person has not abandoned

```

TIME time_advance() const{
    TIME next_internal;
    if (stayingHome)
        next_internal = numeric_limits<TIME>::infinity();
    else {
        int hours = floor(timeR/60);
        next_internal = TIME({hours, timeR - (hours*60)});
    }
    return next_internal;
}

```

**Fig. 11.** Time advance function of the Person class.

```

void confluence_transition(TIME e, make_message_bags<input_ports>::type mbs){
    internal_transition();
    external_transition(TIME(), std::move(mbs));
}

```

**Fig. 12.** Confluence transition function of the Person class.

```

for(int i = 0 ; i < msgBagInToRoom.size(); i++){
    ...
    if (numberPeople < maxOccupancy)
        peopleInRoom.push_back(msgInToRoom);
    else
        strangers = acquaintances = friends = 0;
        for (int j = 0; j < numberPeople; j++){
            for (int l = 0; l < msgInToRoom.relationships.size(); l++){
                if (msgInToRoom.relationships[l].PersonID.compare(ID) == 0){
                    found = true;
                    relationshipType = msgInToRoom.relationships[l].Relationship_type;
                    break;
                }
            }
            assert(found == true);
            if (relationshipType.compare("stranger") == 0) strangers++;
            else if (relationshipType.compare("acquaintance") == 0) acquaintances++;
            else friends++;
            assert(strangers+friends+acquaintances == numberPeople);
            strangersProb =
                relationshipBehaviour.at("stranger").EnterMaxOccRoomProbability*strangers%;
            ...
            uniform_int_distribution<int> probEnterDistribution(0,100);
            r = probEnterDistribution(generator);
            if ((r <= strangersProb)|| (r <= friendsProb)|| (r <= acquaintancesProb))
                peopleInRoom.push_back(msgInToRoom);
            else
                peopleWaitingOutsideRoom.push_back(msgInToRoom);
        }
}

```

**Fig. 13.** Pseudocode to determine if a person will enter a room.

their schedule, we set their next location. We first check if the person is sick but it has not been decided whether they will abandon their schedule. In that case we decide at random to see if a person abandons their schedule by going home. Next we check if the person is travelling for the first time. This is a special case as the simulations start at midnight when the person should have already been at the home location, so the *timeR* state variable is initialized to a smaller value than what would be needed by the *setNextLocation* function to acquire the *LocationPlan* object of their next destination from the schedule stored in their *DecisionMakerBehaviour* object. Next the *travelInfo* object is updated with

their next destination. If the person is becoming sick, we deduct the current time remaining variable from the time until they are sick. If the time until they become sick is less than or equal to zero, then they are now considered to be sick and are no longer becoming sick. Then, if the person's next destination is a travel path then we select which one they will take by comparing the current weather value with the person's weather threshold. If the weather value is greater than or equal to their weather threshold then they will take the safer path, otherwise they will take the riskier path.

The time advance function (Fig. 11) uses an object of the class TIME,

which stores the simulation time in minutes in an hours::minutes::seconds format. The amount of time returned by this function is then the amount of time until the next internal transition fires. So, if the person is currently staying home, they will not be traveling anymore, so the function returns infinity. Otherwise, the time until the next internal transition is updated with the current value of the *timeR* state variable.

The confluence transition function (Fig. 12) triggers when an external transition and an internal transition are triggered at the same time. In the case of this model, the internal transition is set to a default behavior that always fire before the external transition function.

As discussed earlier, the remaining models are constructed similarly using Cadmium. In the rest of the section, we will discuss some interesting aspects of some of the models. To begin, the Room class is mostly passive: it only executes when it receives input from the instances of the Person class. As a result, most of its features are found in the external transition function, including all the calculations that determine the behavior of individuals. The external transition function calculates the CO2 concentration using the equations explained in section 4.3. in 10-minute intervals. A maximum occupancy feature determines if the incoming Person will enter the Room based upon their probability of performing this behavior given their relationships with the people in the room (Fig. 13). First, it checks if the number of people in the Room is below the maximum occupancy limit. The sums of each Person's relationship type with the incoming person are taken, and the percentage of strangers, acquaintances, and friends to the incoming Person in the Room is calculated by dividing the sums by the number of people currently in the Room. Next, the probability of the incoming Person entering a room if they are all strangers is multiplied by the percentage of strangers in the Room. This is repeated with each relationship type in the following code snippet.

The external transition function then iterates through all the messages from the *outFromRoom* input port. The probability of exposure calculation is done by first sorting all the people that the Person leaving interacted with while they were in the room. They are stored in the leaving Person's current relationships map found in the state variable's *relationshipsInRoom* map. This map is then iterated through and each *PersonRelationInfo* struct is stored in a vector according to their relationship type. We then calculate the average probability of the Person leaving having worn a mask or socially distanced themselves, depending on the percentage of their time spent in the Room with strangers, acquaintances, and friends.

To calculate the probability of the Person leaving having been exposed to the virus we first determine what factors to use based on the probabilities of their behaviour and the Room's current circumstances. These factors are integers that range from negative five to positive five, where a positive value increases the odds of exposure, and a negative value decreases the chance of exposure. The values of the factors are chosen based on whether an action should increase or decrease their odds of exposure relative to a rank from one to five. For example, the vaccination factor indicates that the odds of being exposed decrease if the Person is vaccinated, so the value would be negative.

### 6.1. Defining coupled models

To create the coupled models and connect them to form the top model seen in Fig. 6, a C++ script with a "main" function is compiled and executed. The script is available in our repository (Griffith and Baggs, 2022). The script begins by instantiating every instance of the atomic models. The Room atomic models are instantiated by first loading their unique XML files and extracting their attributes using the TinyXML library (Thomason, 2000). The Person atomic models are instantiated from the attributes stored in their XML files in the same way. As all of their attributes are loaded from XML files upon execution, it means that the model does not need to be recompiled with each experiment. Instead, all a person needs to do is change the XML files to suit the parameters of their experiment and run the executable. After the

atomic models have been instantiated, their ports are connected in the manner shown in Fig. 6. The model will then request that the user enter how many hours the simulation will run for in simulated time. The simulation will finish executing and generate logs that show the state of the model at every timestep.

## 7. Studying the spread of COVID-19 at Carleton university

We conducted multiple experiments on the model with COVID-19 and information from Carleton University's campus as a case study. With COVID-19 as our disease being studied, we used an incubation period of two weeks (Lauer et al., 2020). To generate XML files for rooms that have the same characteristics as those found on Carleton's campus, we needed their dimensions and ventilation ratings. The dimensions of 434 rooms from various buildings on Carleton's campus were extracted from a Building Information Modeling (BIM) file. Unfortunately, no ventilation data was obtainable. For simulations where the ventilation rating was not a part of the experiment a ventilation rating of three air changes per hour was used. This value was chosen because it is the recommended ventilation rating for classrooms by the Harvard Schools for Health group (McNeill et al., 2022). These dimensions and ventilation ratings are stored in a csv file that the room generator program reads from when creating room XML files. The remaining parameters are generated in different ways. The social distance threshold is set to one half of the Room's floor space. The maximum occupancy is currently set to three quarters of the social distancing threshold. The mask and social distance factors are random integers from one to two. The vaccinated factor is a randomly generated integer from negative one to negative four. These can be changed easily by the user by editing the relevant part of the *randomGenerator.cpp* file.

To create the XML files for the Person models we chose to replicate the typical schedules of students as they are the largest population on campus. Unfortunately, due to privacy concerns, we could not obtain any examples of real student schedules on which to base our generated schedules. We used typical Carleton University schedules, in which students arrive and depart from classrooms at the same time, each class has a standard duration based upon one of three types (lectures: 1.5 h, labs: 3 h and tutorials: 1 h), and classes only take place between 8:30 am and 9:00 pm. Lectures take place in the largest rooms, labs take place in the second largest rooms and tutorials take place in the smallest rooms. To create the student schedules every room being generated was labeled as a lecture, lab, or tutorial room based on its dimensions. Then, given the hours of operation, each room was assigned as many events that matched their type as could fit into one school day. The generated student schedules were then assigned events at random. In between events, 10 min was given to travel between destinations as per the typical Carleton class schedule. To model the riskier travel path that a student may take, we created an instance of the Room model that is based off the tunnel system that connects every building on Carleton's campus. These tunnels are the paths that students take when they do not want to travel outside to their destination. The XML for the large network of tunnels was arbitrarily assigned large values of 1000 square metres and 4 m for height. A ventilation rating of 3 air changes per hour was also applied to the tunnels.

We conducted preliminary experiments on the effects ventilation quality has on the spread of disease as preliminary reported in (Griffith et al., 2024). In this case study, we present further details on ventilation experiments and experiments on the effects of a policy that mitigates the contact between sick and healthy students.

### 7.1. Analyzing the effect of ventilation policies

This section presents a scenario where we use the simulator to determine whether the quality of ventilation in rooms has an effect on the probability of exposure to COVID-19. The probability of exposure is the chance of a person being exposed to COVID-19 particles. This means

**Table 3**  
Data of Rooms used in Experiments.

Lecture Halls			Laboratories			Tutorials		
Name	Area (m <sup>2</sup> )	Height(m)	Name	Area (m <sup>2</sup> )	Height(m)	Name	Area (m <sup>2</sup> )	Height(m)
38-VS – 3226	313.46	3.048	38-VS – 5112	121.98	3.200	38-VS – 3219	54.77	3.048
38-VS – 2285	185.27	3.200	23-SP – 303	143.65	2.438	38-VS – 3220	45.91	3.048
38-VS – 4217	198.31	3.200	23-SP – 349	122.46	2.438	38-VS – 3221	53.67	3.048
38-VS – 1217	238.53	3.200	23-SP – 280	134.33	2.438	38-VS – 3202	72.91	3.048
12-SC – 246A	286.38	2.438	23-SP – 180	150.28	2.438	38-VS – 3228A	52.48	3.048
12-SC – 214	568.02	2.438	23-SP – 100	120.25	2.438	38-VS – 3228C	65.27	3.048
12-SC – 329	208.14	2.438	23-SP – 128A	112.86	2.438	38-VS – 2104	52.36	3.200
12-SC – 305	323.87	2.438	5-RH – 251	132.82	2.438	38-VS – 2286	54.92	3.200
12-SC – 350A	339.39	2.438	5-RH – 351	132.82	2.438	38-VS – 2202	55.48	3.200
23-SP – 460	209.48	2.438	5-RH – 451	132.84	2.438	38-VS – 2203	53.68	3.200

that if they are exposed, there is a chance for them to become sick, but the actual odds of them becoming sick are closely tied to a person’s medical history and, therefore, cannot be influenced by any COVID-19 safety protocols. Therefore, we only care about a person’s probability of being exposed to the virus. In these experiments, we will simulate 30 rooms where 10 are large enough to be lecture halls, 10 are the size for student’s labs, and 10 rooms are the smallest type for tutorials. 1200 people will be generated along with the rooms, as this is the maximum number of people that can be generated with the chosen rooms. We will run the simulations for two weeks of simulated time as this is the maximum time required for a person to become sick from COVID-19. As a starting point, we will replicate the simulations 5 times to calculate a preliminary confidence interval and decide if more replications are needed, as explained in the analysis of the results. We first use an ACH (Air Changes per Hour) value of 1 for every room, and then an ACH value of 8 for every room to simulate extremely different ventilation rates in the rooms. Only the room representing the tunnels will remain the same as it is constant, and no parts of it are randomly generated. Based on the results from the literature (Arjmandi et al., 2022), we expect that the simulations with an ACH value of 1 will have higher probabilities of exposure and more people becoming sick, while the simulations with an ACH value of 8 will have lower probabilities of exposure and fewer people becoming sick.

Table 3 shows the room size. The rooms shown are divided into an equal number of rooms used for lectures, laboratories, and tutorials to avoid biasing the CO2 concentrations towards larger or smaller rooms.

The remaining simulation parameters are as follows:

- Max Occupancy: ¾ of social distance threshold
- Social Distance Threshold: ½ of room’s area
- Wears Mask Factor: Uniform distribution that is either 1 or 2
- Social Distance Factor: Uniform distribution that it is either 1 or 2
- Vaccinated Factor: Uniform distribution from –1 to –4
- Sick People CO2 Factor: Uniform distribution from 1 to 3
- CO2 Factor Thresholds: 3 values with a uniform distribution between 1000 and 2000 in order from lowest to highest
- CO2 Factors: 1, 2, and 3 assigned to thresholds from lowest to highest
- Respiratory CO2 Increase Per Minute: 340 000

- Room can be Over Capacity: 30 % chance that room can be assigned more students than its max occupancy allows
- Relationships: 20 % are Friends, 30 % are Acquaintances, and 50 % are Strangers
- Sick at Start: 0 % of people are sick at the start
- Vaccinated: 50 % of people are vaccinated
- Wears mask when alone: 50 %
- Weather Threshold: Uniform distribution from 0 to 10
- Probabilities of wearing a mask, social distancing, and entering a room at max occupancy with Friends: 20 %, 20 %, and 80 % respectively
- Probabilities of wearing a mask, social distancing, and entering a room at max occupancy with Acquaintances: 50 % for all three
- Probabilities of wearing a mask, social distancing, and entering a room at max occupancy with Strangers: 80 %, 80 %, and 20 % respectively
- Probability of becoming sick after exposure: 30 %
- Time to become sick: Uniform distribution from 0 min to 10,080 min (two weeks)
- Abandons Schedule and Stays Home: 50 % chance

The parameters’ values were arbitrarily chosen as we could not validate the model, except for the respiratory CO2 increase per minute parameter, which was calculated to be 340000 mg/min as explained in section 4.

The data obtained from each simulation was the CO2 concentration at each half-hour for every room and the probabilities of exposure at each half-hour for every person. 10 people were chosen at random to record their graphs. Their identification numbers are 25, 1, 53, 889, 543, 1099, 765, 999, 812, and 1111.

Table 4 summarizes the statistics from the data obtained in the experiment when ACH is 1. The sample mean, sample variance, and half-width confidence intervals with 95 % confidence were calculated for the probability of exposure values for all 1200 people that were simulated.

Five replications were chosen as a starting point from which the number of replications necessary to have a confidence interval of 1 % for the probability of exposure, and an interval of 10 mg/L for the CO2 values would be determined. Five replications were chosen because the simulations take around 1 h and 30 min to complete, so around 8

**Table 4**  
Data obtained for Experiment ACH = 1: Exposure probability and CO2 concentration.

Replications	Exposure Probability (%)			CO2 Concentration (mg/L)		
	Mean	Variance	Conf. Int.	Mean	Variance	Conf. Int.
E1R1	29.9771	259.0067	0.9115	889.2212	88433.4321	111.0426
E1R2	29.1751	250.9616	0.8972	879.8302	83292.0315	107.7664
E1R3	28.868	237.5691	0.8730	880.686	88638.6487	111.1714
E1R4	30.1844	266.403	0.9244	894.845	87977.8172	110.7562
E1R5	28.5901	229.0607	0.8572	880.77	92257.5773	113.4181
Overall	29.3589	0.48236	0.8624	885.0705	44.4870	8.2817

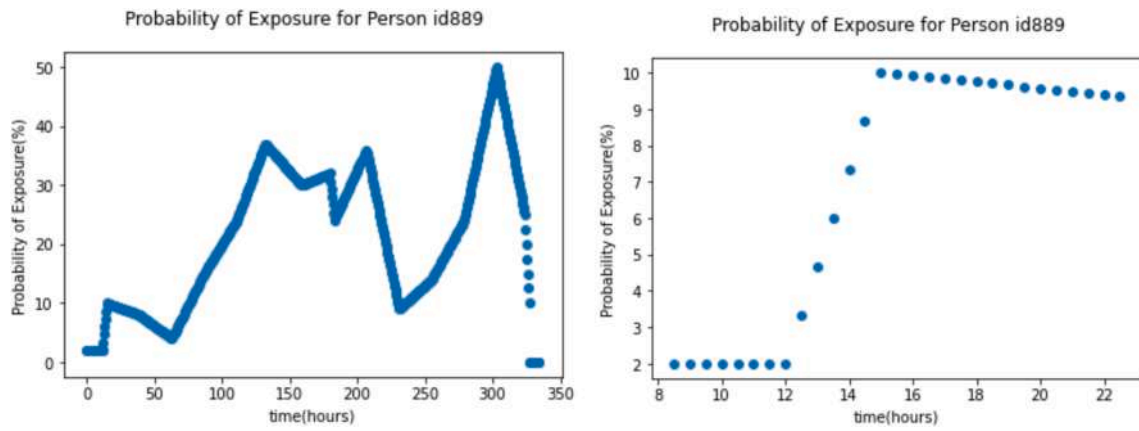


Fig. 14. Probability of Exposure for Person 889. Left: Over two weeks. Right: Over one day.

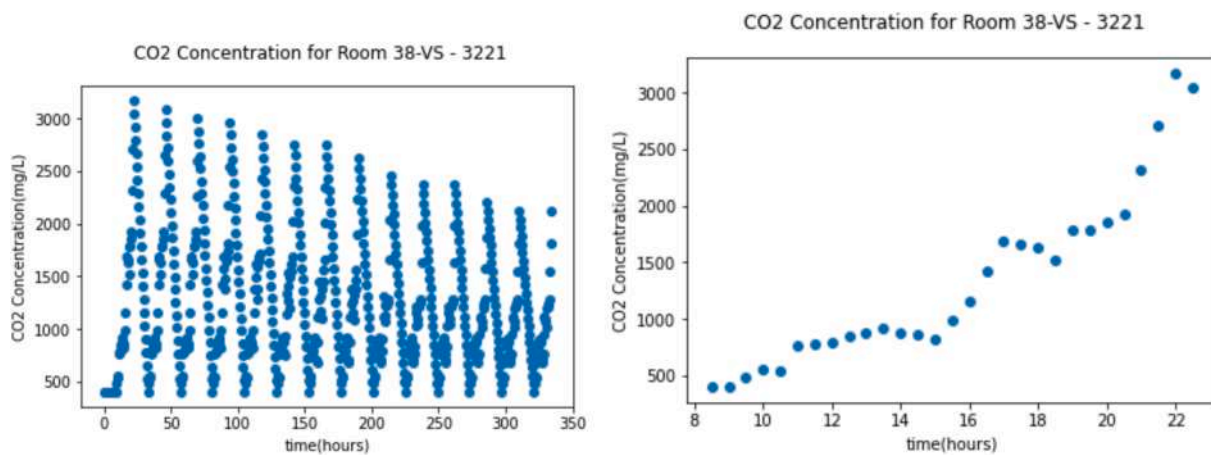


Fig. 15. CO2 Concentration for Room 38-VS – 3221. Left: Over two weeks. Right: Over one day.

**Table 5**  
Data obtained for Experiment ACH = 8: Exposure probability and CO2 concentration.

Replications	Exposure Probability (%)			CO2 Concentration (mg/L)		
	Mean	Variance	Conf. Int.	Mean	Variance	Conf. Int.
E2R1	23.4795	154.3914	0.7037	445.8642	1179.5802	12.8246
E2R2	23.7178	161.6669	0.7201	444.9765	1070.3994	12.2167
E2R3	24.2372	171.1704	0.7410	445.4648	1052.7961	12.1158
E2R4	23.6137	153.7463	0.7023	445.7014	1131.4423	12.5602
E2R5	24.561	178.8042	0.7573	444.4135	1101.6124	12.3936
Overall	23.9218	0.2100562	0.5691	445.2841	0.34897	0.7335

replications can be run in a single day, meaning it takes less than two days to complete both experiments.

After running the five replications, the confidence intervals were already within the target accuracy.

The following graphs were generated from the data obtained in the ACH = 1 experiments. They give some insight into how the simulation progressed when the ventilation rate was set to such a low value.

Fig. 14 (left) shows the probability of exposure for one person throughout the two weeks of replication 5. The reason for changes is unclear. One reason might be off-campus in safer locations or more relaxed schedules on some days of the week. Fig. 14 (right) shows the probability of exposure for the same person over the course of their first day. As the probability of exposure is only calculated when a person leaves a room, there could be very few actual data points if the person visits few rooms. Therefore, we performed linear interpolation between data points.

Fig. 15 (Left) represents the CO2 concentration values recorded for room 38-VS – 3221 at every half hour for two weeks. We can see large erratic spikes followed by linear dips; these represent the day when people are using the room and breathing CO2 into the air, but with poor ventilation, it only keeps increasing throughout the day. Then, once the room is empty for the night, the ventilation can bring it back down to normal levels. The rise appears to be erratic, but it is not. This is because the CO2 concentration is calculated whenever a person enters or leaves the room using the amount of time that has passed since either of those events occurred. This means that the CO2 concentration appears to suddenly jump at times because a class is over, and the new CO2 concentration has been calculated considering the 90-minute lecture or 180-minute lab that has just finished.

Fig. 15 (Right) shows the CO2 concentration values taken for room 38-VS – 3221 over the course of one day. Here, it can be seen how the CO2 concentration quickly rises throughout the day because of the poor

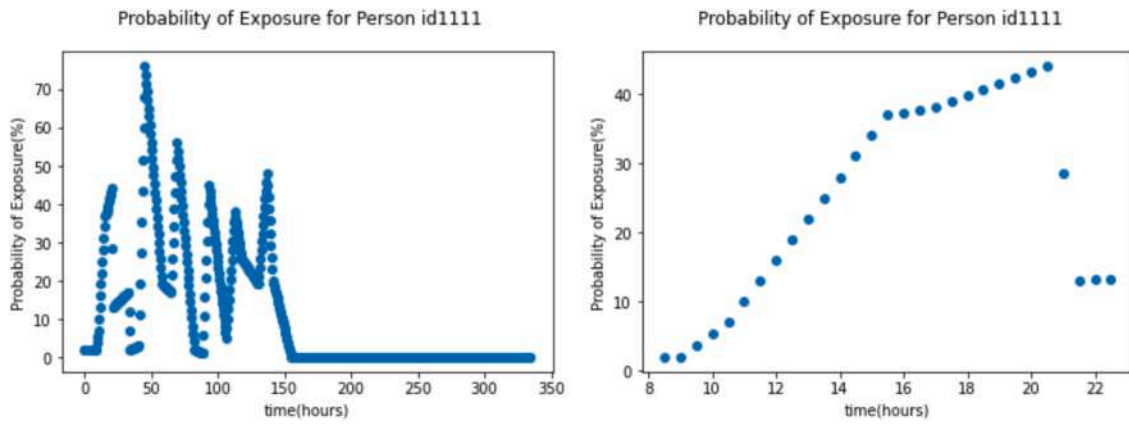


Fig. 16. Probability of exposure for Person 1111. Left: Over 2 weeks. Right: Over one day.

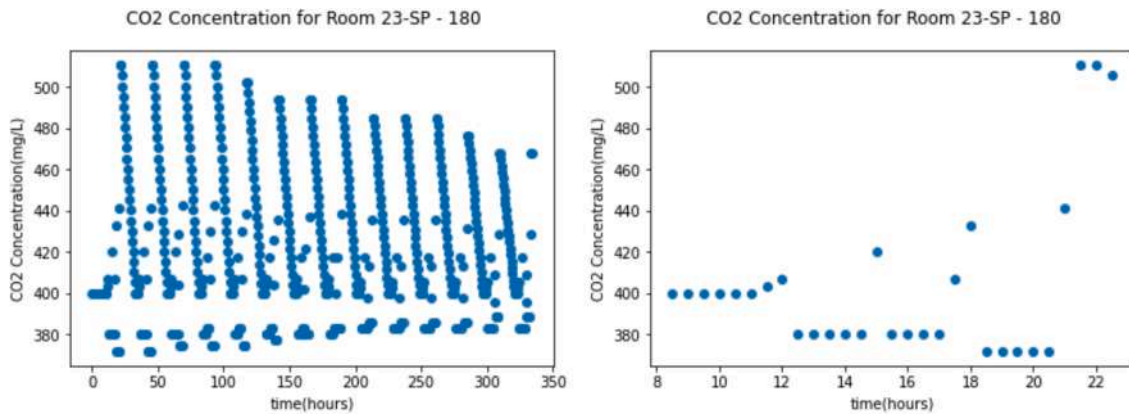


Fig. 17. CO2 Concentration for room 23-SP – 180 Left: Over 2 weeks. Right: Over one day.

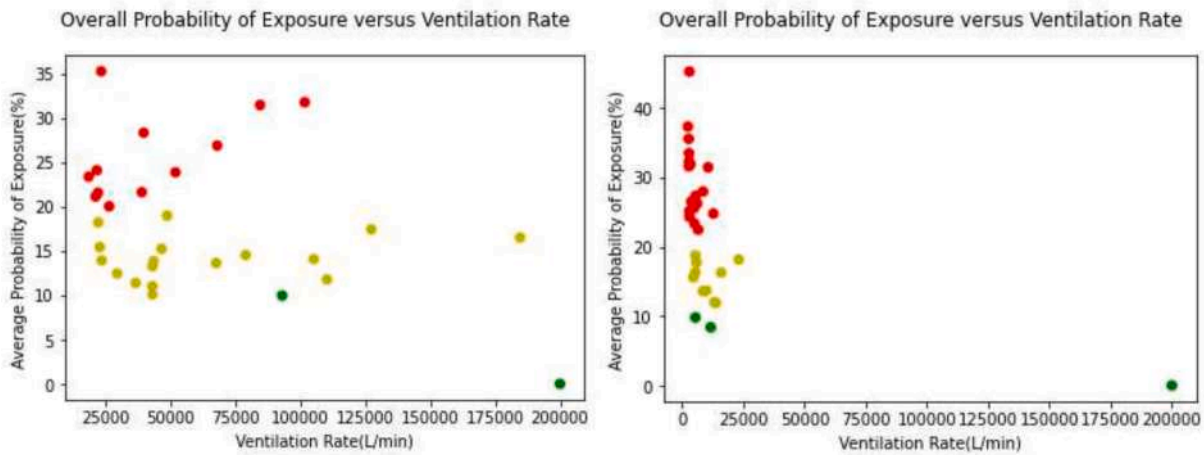


Fig. 18. (On Left): Average Probability of exposure generated in each room versus ventilation rate per room. Left: ACH value = 8 Right: ACH value = 1.

ventilation. Some data had to be linearly interpolated, but far fewer than the values required for the probability of exposure.

Table 5 summarizes the statistics from the experiment when ACH is 8. We obtained the same data and statistics as in the previous experiment and found that the number of replications was enough for a 95 % confidence interval. As we can see from the data, the variance of the CO2 values is smaller than in the previous case, as the ACH value is much higher. This means that the CO2 values will fluctuate far less than they did when an ACH value of 1 was used, as the CO2 peak will be lower.

The following graphs were generated from the data obtained with an ACH value of 8. They give some insight into how the simulation progressed when the ventilation rate was high.

Fig. 16 (left) shows the probability of exposure for one person when ACH was set to 8 over the entire two weeks of the simulation. The graph flattens at 0 % around halfway through the simulation. This is because the person became sick, and at that point, we are no longer interested in recording their probability of becoming exposed as they are already sick. Fig. 16 (right) shows the probability of exposure for one person when



**Table 6**

Data and statistics of the probability of exposure with a 0% probability of abandoning the schedule when sick.

Replications	Mean	Sample Variance	Conf. Int. @ 95 %
E3R1	23.8803	164.4753	0.726350829
E3R2	25.9279	203.2739	0.807489535
E3R3	24.4434	176.0343	0.751440748
E3R4	24.9802	186.1616	0.772753791
E3R5	25.3637	193.2622	0.787353119
Overall	24.9191	0.6311189	0.98641506

ACH is set to 8 over the course of one day. We can see that it increases steadily and then drops off as they leave the campus to go home.

Fig. 17 (left) shows the CO2 concentration for room 23-SP – 180 with ACH = 8 over the entire two weeks of the simulation. The highest points of CO2 concentration decrease as the week progresses. This is because more people are becoming sick and staying home, so there are fewer people contributing to the CO2 concentration of the room as the days progress. Fig. 17 (right) shows the CO2 concentration for room 23-SP –180 with ACH = 8 over the course of one day. While the CO2 concentration keeps dropping suddenly, as it is calculated only when people leave the room and not periodically, it is changing far less than the CO2 concentration of rooms when ACH = 1 because it has a much larger ventilation rate.

Fig. 18 shows every room’s ventilation rate compared to its average probability of exposure values generated. The graph on the left is for the rooms where the ACH value equaled 8, and the graph on the right is for the rooms where the ACH value equaled 1. The dot in the bottom right is the same for both graphs, and it represents the tunnels on Carleton’s campus, which are very large and well-ventilated, so the odds of becoming exposed to COVID-19 remain low. It can be seen, however, that when more points have a lower ventilation rate, they tend to have a higher average probability of exposure. The rooms in the safest group are below 10 (dark green), rooms in the somewhat safe group are less than 20 but greater than 10 (yellow), and the rooms in the dangerous group are greater than 20 (red).

The results of the experiment where ACH = 1 show an overall average probability of exposure equal to  $29.3589 \pm 0.8624$  % and an overall average CO2 concentration of  $885.0705 \pm 8.2817$  mg/L. The results of the experiment where ACH = 8 show an overall average probability of exposure equal to  $23.9218 \pm 0.5691$  % and an overall average CO2 concentration of  $445.2841 \pm 0.7335$  mg/L.

These averages show that when the ventilation of all rooms on campus are improved by 7 air changes per hour, we would see the average probability of becoming exposed to COVID reduce by  $5.4371 \pm 1.0346$  %. The results, therefore, prove that implementing a policy where the ventilation rates of rooms on Carleton’s campus are standardized at a high ACH value would improve safety and limit the spread of the virus.

**Table 7**

Data and statistics of the probability of exposure with a 100% probability of abandoning the schedule when sick.

Experiments	Mean	Sample Variance	Conf. Int. @ 95 %
E4R1	20.8052	109.084	0.591529789
E4R2	20.108	97.3766	0.558886213
E4R3	21.3385	113.3007	0.602854336
E4R4	21.4731	124.4235	0.631752981
Overall	20.9312	0.384361	0.986508367

7.2. Studying the stay at home policy

In this section we simulate a COVID-19 safety policy that does not allow sick people on Carleton’s campus. We use the same rooms, the same simulation parameters, and the same simulated students. The only difference is that the ACH value will be set to a middle value of 4. In the first experiment, we set the probability of a person abandoning their schedule and staying home to 0 % and in the second one, to 100 %. In both cases, the probability of exposure will be examined.

Table 6 summarizes the statistics from the experiment when a 0 % probability of abandoning the schedule is used. We obtained the probability of exposure and calculated the mean, variance, and 95 % confidence interval. As in the previous scenario, we found that our starting 5 replications were enough for a 95 % confidence interval below 1 %.

The following graphs were generated from the data obtained in the experiments where the stay-home-when-sick policy was not active. They give some insight into how the simulation progressed if the sick students were allowed onto the campus.

Fig. 19 (left) represents the probability of exposure of one person when the stay-home-when-sick policy is not in effect for the entire two weeks of the simulation. It peaks after the halfway point and then drops significantly, though the cause is unclear. Fig. 19 (right) represents the probability of exposure of one person when the stay-home-when-sick policy is not in effect over the course of one day. It starts high as that is the last exposure value they received from visiting a room and then drops as they receive a lower value during their travels.

Table 7 summarizes the statistics from the experiment when a 100 % probability of abandoning the schedule is used. We obtained the probability of exposure and calculated the mean, variance, and the 95 % confidence interval below 1 %. We first used 3 replications, and we found that they were not enough for a 95 % confidence interval below 1 %. Thus, we increased the value to 4, and we found that they satisfied the 95 % Confidence Interval requirement.

The following graphs were generated from the data obtained in the experiments where the stay-home-when-sick policy was active. They give some insight into how the simulation progressed when the sick students are not allowed onto the campus.

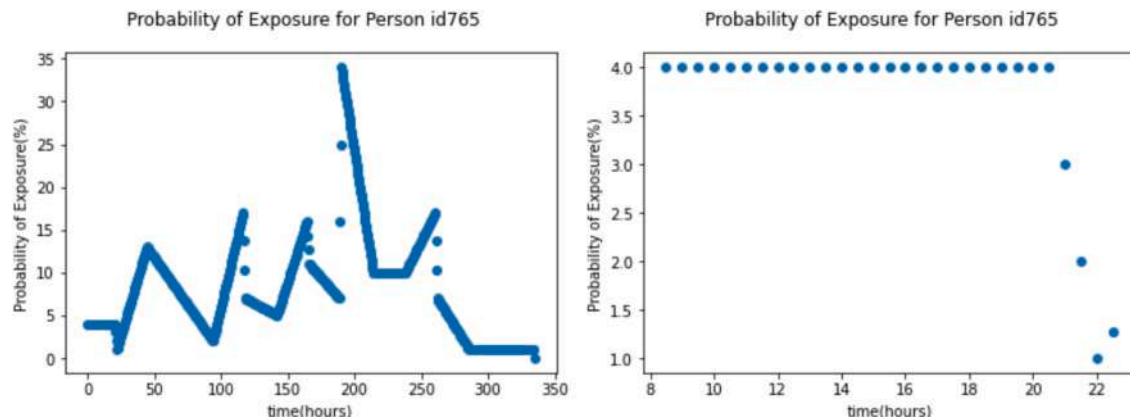


Fig. 19. Probability of Exposure for Person 765. Left: Over 2 weeks. Right: Over one day.

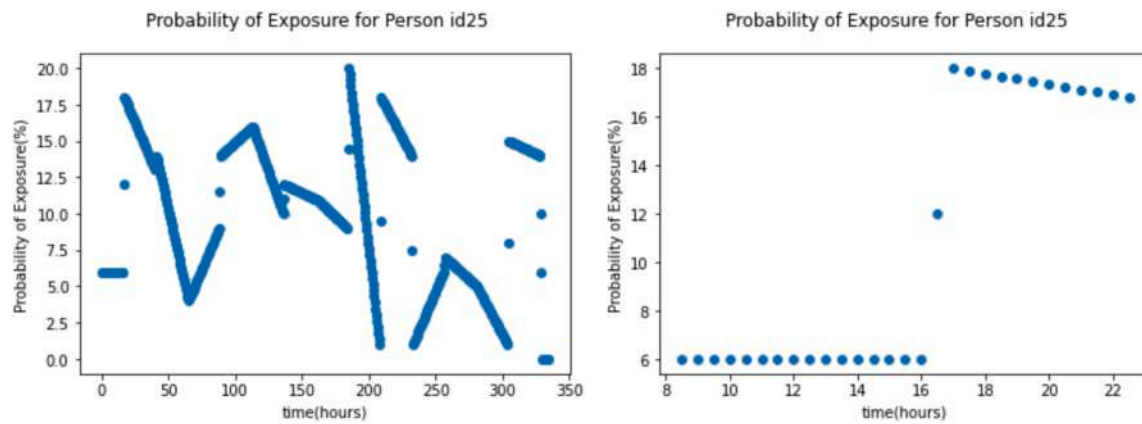


Fig. 20. Probability of exposure for Person 25. Left: Over 2 weeks. Right: Over one day.

Fig. 20 (left) represents the probability of exposure of a person over the entire two weeks of the simulation. We can see that it never goes far above 20 % which is consistent with the overall average found for the experiment. Fig. 20 (right) represents the probability of exposure for replication of a person over the course of one day. In this case, they only had one or two events in their schedule as it shows little variation, and most values have been linearly interpolated.

The results of not applying the stay-home-when-sick policy show an overall average probability of exposure equal to  $24.9191 \pm 0.9864$  %, while the results of applying the policy show an overall average probability of exposure equal to  $20.9312 \pm 0.9865$  %.

This means that when a policy of not allowing sick people on Carleton's campus is applied, there is, on average, a reduction of  $3.9879 \pm 1.3108$  %. This result shows the benefits of applying the policy. However, the simulation results show that it may be less beneficial than what one can a priori expect.

### 7.3. Discussion

The simulation results obtained by following the proposed models and methods can have many implications for policymakers of workplaces, depending on the workplace characteristics being studied.

For example, in our case study, students followed a stay-at-home policy if they became sick with COVID-19. The simulation results from these experiments showed that the overall average probability of exposure to COVID-19 decreases when this policy is in effect. This implies that fewer students could become sick with COVID-19 if this policy is enforced. A policymaker for Carleton University could enact this policy based on those positive results, or they may decide that the benefits of this policy do not outweigh the cost of enforcing it properly (not only in terms of costs but in terms of social aspects such as student isolation or missed classes that could affect the student's performance). In those situations, where the simulation results show that the application of a policy results in the desired outcome but has associated costs, further analysis is needed. For example, the policymaker may conduct more experiments with the same policy, but different percentages of sick students are allowed to go to campus and evaluate the trade-offs in those situations. Additionally, the policymaker may also raise other concerns, such as compliance fatigue. Thus, we would need to conduct simulations where the percentage of students who follow the policy varies over time. These additional experiments could assist policymakers by allowing them to assess various levels of compliance fatigue, thereby informing more effective decision-making.

Let us assume a scenario where the compliance fatigue experiments prove that the policy is not worth the cost and implementation effort. In that case, the policymaker could consider different policies to help reduce the average probability of exposure to COVID-19, and those can be evaluated through our simulation model. For example, in our case

study, we conducted experiments that compare the average probability of exposure when each room has the same low- or high- quality ventilation rating. These simulation results showed that a high-quality ventilation rating in each room would reduce the average probability of exposure to COVID-19. This implies that policymakers could reduce the number of students who become sick with COVID-19 by enacting a policy that sets a standard high-quality ventilation rating for every room. The policymaker could consider renovation costs by determining the optimal cost-to-benefit ratio by conducting further experiments with different ventilation ratings. In this way, the policymaker could use the implications of the simulation results to enact policies that benefit the health of the employees in a workplace without straining the business's funds or increasing the employees' compliance fatigue.

We acknowledge that policymakers will always have a trade-off analysis situation scenario, and iterations over the results to test further policies may be needed. Some of the interactions may even include upgrades to the model.

## 8. Conclusions

Carleton University's campus was underused during the middle of the COVID-19 pandemic because it was too dangerous for students and faculty to attend it safely. However, it can be made a safer place for people to use the campus facilities. This can be done by implementing COVID-19 safety policies that improve the safety of those who visit the campus. Though, how do we decide which safety policy to use over another? We can not test them with real people on the real campus. Therefore, our solution is to create a simulator that can determine the probability of people becoming exposed to COVID-19 given many possible factors. To do so, we built a model using discrete-event systems specifications and an architecture to model diffusion processes in multiplex dynamic networks. We use agents to describe the behavior of individuals following or not following safety policies whose interactions are dynamically created direct links across a network of rooms that represent real rooms on Carleton's campus to model how COVID-19 might spread across those links and expose people to the virus. The proposed model has over 30 ABM that generate over 200 DEVS atomic models (one of such models has been included as an example to demonstrate the functionality of the proposed architecture). All the models and their implementation are available in (Griffith and Baggs, 2022).

The simulated results showed how ventilation quality impacts the probability of a person becoming exposed to COVID-19. This was done by running simulations where every room had an ACH value of 1, so they have very poor ventilation and running simulations where every room had excellent ventilation with an ACH value of 8. The results showed that poor ventilation can lead to an increased average probability of exposure over 5 %. We also used a policy where people who are sick with

COVID-19 are not allowed to be on Carleton's campus. This was done by running experiments where everyone who became sick would not stay home, and by running other simulations where everyone who became sick would stay home. The results showed that denying sick people access to Carleton's campus reduced the average probability of being exposed to COVID-19 by around 4 %.

The case study's accuracy is limited by many assumptions. These include, among others, how students follow the same schedule every day, how the ventilation rating is the same for every room, and how the ventilation rating does not change over time. These limit the accuracy of the case study's results by oversimplifying the real-world phenomena we are simulating but allow us to exemplify better the use of the methodology to analyze and study disease spread scenarios in a workplace considering human behavior and building characteristics. In real life, a student's schedule changes every day and the ventilation rating of one room is likely different from the ventilation ratings of neighboring rooms. However, these assumptions (made due to limited available data) can be revisited to reflect real-world phenomena more accurately. For example, to make a student's behaviour vary over the course of a week, we just change their associated XML file. To make many students follow different behavioural patterns, then the script that randomly generates student XML files can be edited in such a way that different number of students follow one pattern or more. To change any room's ventilation rate to a different value or to a distribution of values, we just need to change its associated XML file. Using this method, the assumptions can be easily modified or eliminated as more information is obtained about the workplace being modeled.

Adapting this discrete-event methodology to a different workplace would follow a process of information gathering and XML file generation. All the models developed stay the same. The most essential information would be the dimensions and ventilation rating of each room being included, as well as an average employee's daily schedule. The amount of data collected will determine whether some XML files need to be randomly generated with fixed percentages based on average attributes or XML files can be generated with real information from a database. These XML files contain every unique characteristic of the workplace being adapted. For example, in the case study with Carleton University's campus, we included the unique tunnel system by treating it as a very large room. The same could be done with the hallways on each floor of an office building represented as an XML file, where an employee's schedule in their XML file may represent them regularly traversing crowded hallways as they leave their office to attend a meeting elsewhere. A conventional office building where employees travel through crowded hallways could have each floor's network of hallways represented by one large room's XML file. Then each hallway network would be a room that an employee spends a period of time in before transitioning to a different room in the building.

To scale the simulations down to a small section of a workplace or scale it up to a large workplace, a user only needs to create the minimum necessary room and behavioural XML files. However, a workplace could easily have thousands of rooms and employees, which increases the computational load of the simulations being executed. To reduce this problem, you can make reasonable assumptions, such as not including rooms where people do not congregate, like storage rooms or bathrooms. The ADMP methodology proposed in (Ruiz-Martin et al., 2018) has, therefore, been successfully adapted to study the spread of airborne diseases in workplaces. We have also included all the models needed in our GitHub repository (Griffith and Baggs, 2022). Thus, when all the necessary XML files are generated with the unique characteristics of a workplace, other settings and institutions can be studied. Additionally, the models and the XML files generated are easy to adapt to include other features that policymakers may be interested in analyzing, such as the odds of employees having passive immunity, the use of plastic barriers to separate people, or if some neighboring rooms have more airflow between them.

## CRediT authorship contribution statement

**Hazel Griffith:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Conceptualization. **Cristina Ruiz-Martin:** Writing – review & editing, Supervision, Methodology, Investigation. **Gabriel Wainer:** Writing – review & editing, Supervision, Methodology, Funding acquisition.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper. This research has been funded by NSERC.

## Data availability

There is a link to the model implementation in the paper (our github repository)

## References

- Abdollahi, E., Haworth-Brockman, M., Keynan, Y., et al. (2020). Simulating the effect of school closure during COVID-19 outbreaks in Ontario, Canada. *BMC Medicine*, 18, 230. <https://doi.org/10.1186/s12916-020-01705-8>
- Aganovic, A., Bi, Y., Cao, G., et al. (2021). Estimating the impact of indoor relative humidity on SARS-CoV-2 airborne transmission risk using a new modification of the Wells-Riley model. *Building and Environment*, 205. <https://doi.org/10.1016/j.buildenv.2021.108278>
- Aguilar, J., Bassolas, A., G. Ghoshal, S. Hazarie, A. Kirkley, M. Mazzoli, S. Meloni, S. Mimar, V. Nicosia, J. Ramasco, A. Sadleir, "Impact of urban structure on COVID-19 spread," arXiv:2007.15367, (2020) <https://arxiv.org/abs/2007.15367v1>.
- Akbarpour, M., Cook, C., Marzuoli, A., et al. (2020). Socioeconomic network heterogeneity and pandemic policy response. *NBER Working Paper No. 27374*. [https://www.nber.org/system/files/working\\_papers/w27374/w27374.pdf](https://www.nber.org/system/files/working_papers/w27374/w27374.pdf).
- Altamimi, T., Khalil, H., Rajus, V., Carriere, R., & Wainer, G. (2021). Cell-DEVS models with BIM integration for airborne transmission of COVID-19 indoors. *Presented at SimAUD* [Online]. Available: <https://cell-devs-02.sce.carleton.ca/publications/2021/AKSCW21/Viral%20particles%20spread%20indoor%20models-SimAud-98.pdf>.
- Arjmandi, H., Amini, R., Kashfi, M., Alexander Abikenari, M., & Davani, A. (2022). Minimizing the COVID-19 spread in hospitals through optimization of ventilation systems. *Physics of Fluids*, 34(3). <https://doi.org/10.1063/5.0081291>, 037103.
- Azhar, S., Khalfan, M., & Maqsood, T. (2012). Building Information Modeling (BIM): Now and beyond. *The Australasian Journal of Construction Economics and Building*, 12 (4), 15–28. <https://search.informit.org/doi/10.3316/informit.013120167780649>.
- Bahl, R., Eikmeier, N., Fraser, A., Junge, M., Keesing, F., et al. (2021). Modeling COVID-19 spread in small colleges. *PLoS One*, 16(8). <https://doi.org/10.1371/journal.pone.0255654>. e0255654.
- Barman, M., Nayak, S., Yadav, M. K., et al. (2020). Modeling control, lockdown & exit strategies for COVID-19 pandemic in India. *medRxiv*. <https://arxiv.org/abs/2007.07988>.
- Borowiak, M. Ning, F. Pei, J. Zhao, S. Tung, H. Durrett, R. Controlling the spread of COVID-19 on college campuses, arXiv:2008.07293, doi: 10.48550/arXiv.2008.07293.
- Cárdenas, R., Henares, K., Ruiz-Martin, C., & Wainer, G. (2021). Cell-DEVS Models for the Spread of COVID-19. In T. M. Gwizdalla, L. Manzoni, G. C. Sirakoulis, S. Bandini, & K. Podlaski (Eds.), *Cellular Automata. ACR1 2020. Lecture Notes in Computer Science*. Cham: Springer. doi: 10.1007/978-3-030-69480-7\_24.
- D'Orazio, M., Bernardini, G., & Quagliarini, E. (2021). A probabilistic model to evaluate the effectiveness of main solutions to COVID-19 spreading in university buildings according to proximity and time-based consolidated criteria. *Building Simulation*, 14, 1795–1809. <https://doi.org/10.1007/s12273-021-0770-2>
- Davidson, G., Fahlman, A., Mereu, E., et al. (2023). A methodological approach for modeling the spread of disease using geographical discrete-event spatial models. - *Online Simulation*. <https://doi.org/10.1177/00375497231152458>.
- Fahlman, A., Ruiz-Martin, C., Wainer, G., Dobias, P., & Rempel, M. (2021). Extended Compartmental Model of Covid-19: A Cell-DEVS Definition. In *2021 IEEE/ACM 25th International Symposium on Distributed Simulation and Real Time Applications (DS-RT) Valencia, Spain* (pp. 1–8), 10.1109/DS-RT52167.2021.9576127. <https://ieeexplore.ieee.org/abstract/document/9576127>.
- Fahlman, Aidan, & Wainer, Gabriel (2024). Seaird Model to Simulate the Impact of Human Behaviors. In *Proceedings of the Winter Simulation Conference (WSC '23)* (pp. 1172–1183). IEEE Press.
- Ghaffarzagdegan, N. (2021). Simulation-based what-if analysis for controlling the spread of Covid-19 in universities. *PLoS One*, 16(2). <https://doi.org/10.1371/journal.pone.0246323>. e0246323.

- Gómez, S., Díaz-Guilera, A., Gómez-Gardeñes, J., Pérez-Vicente, C. J., Moreno, Y., & Arenas, A. (2013). Diffusion dynamics on multiplex networks. *Physical Review Letters*, 110(2), Article 028701.
- H. Griffith, (Jan. 20, 2022a), "Person3.xml" (1.0) [XML file]. [https://github.com/HazelGriffith/COVID\\_Campus\\_Simulation](https://github.com/HazelGriffith/COVID_Campus_Simulation).
- H. Griffith, (Jan. 20, 2022b), "5,RH - 110C.xml" (1.0) [XML file] [https://github.com/HazelGriffith/COVID\\_Campus\\_Simulation](https://github.com/HazelGriffith/COVID_Campus_Simulation).
- H. Griffith, B. Baggs (2022) COVID-Campus-Simulation [Source code]. [https://github.com/SimulationEverywhere-Models/COVID\\_Campus\\_Simulation](https://github.com/SimulationEverywhere-Models/COVID_Campus_Simulation).
- Griffith, H., Ruiz-Martin, C., Wainer, G. (2024). Discrete-Event Modeling of Human Behavior for Spread of Diseases on University Campuses. In: Mujica Mota, M., Scala, P. (eds) Simulation for a Sustainable Future. EUROSIM 2023. Communications in Computer and Information Science, vol 2032. Springer, Cham. [https://doi.org/10.1007/978-3-031-68435-7\\_8](https://doi.org/10.1007/978-3-031-68435-7_8).
- Gressman, P., & Peck, J. S. (2020). COVID-19 in a university environment. *Mathematical Biosciences*, 328. <https://doi.org/10.1016/j.mbs.2020.108436>, 108436, ISSN 0025-5564.
- Hallet, S., Toro, F., & Ashurst, J. V. (2021). *Physiology, tidal volume*. Statpearls Publishing Accessed on: Sept. 30, 2021. [Online] Available at: <https://pubmed.ncbi.nlm.nih.gov/29494108/>.
- Hekmati, A., Luhar, M., Krishnamachari, B., & Mataric, M. (2021). Simulation-based analysis of COVID-19 spread through classroom transmission on a university campus. In *2021 IEEE International Conference on Communications Workshops (ICC Workshops), Montreal, QC, Canada* (pp. 1–6). doi: 10.1109/ICCWorkshops50388.2021.9473803 Available: <https://ieeexplore.ieee.org/abstract/document/9473803>.
- Hekmati, A., Luhar, M., Krishnamachari, B., & Mataric, M. (2022). Simulating COVID-19 classroom transmission on a university campus. *Proceedings of the National Academy of Sciences*, 119(22). <https://doi.org/10.1073/pnas.2116165119>
- Hethcote, H. W. (2020). Mathematics of infectious diseases. *SIAM Review*, 42, 599–653.
- Hussein, T., Löndahl, J., Thuresson, S., Alsveld, M., Al-Hunaiti, A., Saksela, K., Aqel, H., Junninen, H., Mahura, A., & Kulmala, M. (2021). Indoor model simulation for COVID-19 transport and exposure. *International Journal of Environmental Research and Public Health*, 18(6), 2927. <https://doi.org/10.3390/ijerph18062927>
- Lauer, Stephen A., Grantz, Kyra H., Bi, Qifang, et al. (2020). The incubation period of coronavirus disease 2019 (COVID-19) from publicly reported confirmed cases: Estimation and application. *Annals of Internal Medicine*, 172, 577–582 [Epub 10 March 2020]. doi:10.7326/M20-0504.
- Li, X., Lester, D., & Rosengarten, G. (2022). A spatiotemporally resolved infection risk model for airborne transmission of COVID-19 variants in indoor spaces. *Science of The Total Environment*, 812. <https://doi.org/10.1016/j.scitotenv.2021.152592>, 152592, ISSN 0048-9697.
- Lin, B., Chen, Y., Hung, C., et al. (2021). The data forecast in COVID-19 model with applications to US, South Korea, Brazil, India, Russia, and Italy. *ArXiv*. <https://arxiv.org/abs/2011.04738>.
- Liu, H., He, S., Shen, L., & Hong, J. (2021). Simulation-based study of COVID-19 outbreak associated with air-conditioning in a restaurant. *Physics of Fluids*, 33(2). <https://doi.org/10.1063/5.0040188>
- Lopman, B. Liu, C. Le Guillou, A. et al. A model of COVID-19 transmission and control on university campuses. medRxiv 2020.06.23.20138677; doi: 10.1101/2020.06.23.20138677.
- Lorig, F., Johansson, E., & Davidsson, P. (2021). Agent-based social simulation of the covid-19 pandemic: A systematic review. *Journal of Artificial Societies and Social Simulation*, 24(3), 5. <https://doi.org/10.18564/jasss.4601>
- Macal, C. M., & North, M. J. (2005). Tutorial on agent-based modeling and simulation. In *Proceedings of the Winter Simulation Conference, 2005., Orlando, FL, USA* (p. 14). doi: 10.1109/WSC.2005.1574234 Available: <https://ieeexplore.ieee.org/abstract/document/1574234>.
- McNeill, V. F., Corsi, R., Huffman, J. A., King, C., Klein, R., Lamore, M., et al. (2022). room-level ventilation in schools and universities. *Atmospheric Environment: X*, 13. <https://doi.org/10.1016/j.aeaoa.2022.100152>, 100152, ISSN 2590-1621.
- Muller, K., & Muller, P. (2021). Mathematical modelling of the spread of COVID-19 on a university campus. *Infectious Disease Modelling*, 6, 1025–1045. <https://doi.org/10.1016/j.idm.2021.08.004>. ISSN 2468-0427.
- Peng, Z., & Jimenez, J. L. (2021). *Environmental Science & Technology Letters*, 8(5), 392–397. <https://doi.org/10.1021/acs.estlett.1c00183>
- Possik, J., et al. (2023). An agent-based modeling and virtual reality application using distributed simulation: Case of a COVID-19 intensive care unit. *IEEE Transactions on Engineering Management*, 70(8), 2931–2943. <https://doi.org/10.1109/TEM.2022.3195813>. <https://ieeexplore.ieee.org/document/9861777/>.
- Priban, I. P. (1963). An analysis of some short-term patterns of breathing in man at rest. *The Journal of Physiology*, 166(3), 425. <https://doi.org/10.1113/jphysiol.1963.sp007114>
- Ross, R. (1916). An application of the theory of probabilities to the study of a priori pathometry— Part I. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 92, 204–230.
- Ross, R., & Hudson, H. (1917). An application of the theory of probabilities to the study of a priori pathometry—Part II. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 93, 212–225.
- C. Ruiz-Martin, G. Wainer (2020), "Cadmium. A tool for DEVS modeling and simulation. User's guide," Dept of Systems and Computer Engineering, Carleton University, [Online] Available: <https://www.sce.carleton.ca/courses/sysc-5104/lib/exe/fetch.php?media=cadmium.pdf>.
- Ruiz-Martin, C., Wainer, G., & Lopez-Paredes, A. (2018). Discrete-event modeling and simulation of diffusion processes in multiplex networks. *ACM Transactions on Modeling and Computer Simulation*, 31(1), 1–32, 6.
- Satish, U., Mendell, M. J., Shekhar, K., Hotchi, T., Sullivan, D., Streufert, S., & Fisk, W. J. (2012). Is CO2 an indoor pollutant? Direct effects of low-to-moderate CO2 concentrations on human decision-making performance. Available at: *Environmental Health Perspectives*, 120(12), Dec <https://ehp.niehs.nih.gov/doi/full/10.1289/ehp.1104789>.
- Sen, D., & Sen, D. (2021). Use of a modified SIRD model to analyze COVID-19 data. *Industrial and Engineering Chemistry Research*, 60, 4251–4260.
- Sidhardhan, S., & Das, D. (2021). Indoor Carbon dioxide (CO2) level control using Wearable smart watches over a wireless channel. In *2021 International Conference on Computer Communication and Informatics (ICCCI)* (pp. 1–4), 10.1109/ICCCI50826.2021.9402692.
- Stephens, B. (2013). HVAC filtration and the Wells-Riley approach to assessing risks of infectious airborne diseases. *NAFA* Accessed: Feb. 7, 2024. [Online] Available: [https://built-envi.com/publications/nafa\\_iit\\_wellsriley%20-%20FINAL.pdf](https://built-envi.com/publications/nafa_iit_wellsriley%20-%20FINAL.pdf).
- Thomason, L (2000) TinyXML [Source code]. <https://sourceforge.net/projects/tinyxml/>.
- Vecherin, S., Chang, D., Wells, E., et al. (2022). Assessment of the COVID-19 infection risk at a workplace through stochastic microexposure modeling. *Journal of Exposure Science & Environmental Epidemiology*, 32, 712–719. <https://doi.org/10.1038/s41370-022-00411-2>
- Wainer, G. A. (2009). *Discrete-Event Modeling and Simulation: A Practitioner's Approach* (1st. ed.). USA: CRC Press Inc.
- Xie, G. (2020). A novel Monte Carlo simulation procedure for modelling COVID-19 spread over time. *Scientific Reports*, 10(13120). Available: <https://www.nature.com/articles/s41598-020-70091-1>.
- Zeigler, B. P., Muzy, A., & Kofman, E. (2019). *Theory of Modeling and Simulation* (3rd ed.). Academic Press.
- Zhang, R., Li, Y., Zhang, A., et al. (2020). Identifying airborne transmission as the dominant route for the spread of COVID-19. *PNAS*, 117(26). <https://doi.org/10.1073/pnas.2009637117>
- Zhou, Y., Li, L., Ghasemi, Y., Kallagudde, R., Goyal, K., & Thakur, D. (2021). An agent-based model for simulating COVID-19 transmissions on university campus and its implications on mitigation interventions: A case study. *Information Discovery and Delivery*, 49(3), 216–224. <https://doi.org/10.1108/IDD-12-2020-0154>