CD^2: AN AUTOMATION TOOL FOR CELL-DEVS CO$_2$ DIFFUSION MODELS

Hoda Khalil
Gabriel Wainer

Department of Systems and Computer Engineering
Carleton University
1125 Colonel by Drive, Ottawa, ON, Canada
hodakhalil@cmail.carleton.ca, gwainer@sce.carleton.ca

ABSTRACT

Measuring Carbon Dioxide and studying its diffusion indoors has many applications, which include, but not limited to, maintaining air quality, conserving energy, and minimizing viral infections. On the one hand, many of the experiments needed to conduct studies, like the indoor diffusion of Carbon Dioxide, are complex to implement. On the other hand, adjusting each model manually to mimic the studied space is difficult and prone to error. We propose an automation tool for modeling, simulating, and visualizing Carbon Dioxide. The tool automates the simulation of Cellular Discrete Event Simulation models of Carbon Dioxide dispersion indoors. We discuss the tool's applications, the software architecture, related tools, and a case study. The case study compares simulations of a closed space with different configurations. The results show how such configurations affect Carbon Dioxide concentration.

Keywords: Carbon Dioxide, Cell-DEVS, automation, indoors models.

1 INTRODUCTION

Designers of modern buildings strive to reduce energy consumption while maintaining a healthy and comfortable indoor environment for the occupants of the buildings. An important indicator of the quality of indoor environments is the concentration level of Carbon Dioxide (CO$_2$). Measuring and understanding the dispersion of CO$_2$ is not only essential for maintaining an acceptable comfort level for occupants, but it is also useful for occupants' detection and demand control of the buildings' services. However, studying the dispersion of CO$_2$ indoors is not attainable by physical experiments as the gas behavior changes significantly depending on the configuration of the space (e.g., dimensions of the space and presence of windows or vents). Therefore, modeling and simulation (M&S) of CO$_2$ diffusion indoors is a viable alternative for architects.

Cellular Discrete EVent Specifications (Cell-DEVS) models have been created by researchers to study the diffusion of carbon dioxide (CO$_2$) indoors. The models have been proven effective for understanding several effects of CO$_2$ accumulation in different rooms (Khalil, Wainer, and Dunnigan 2020; Khalil and Wainer 2020b). An example of the application of such models is understanding the effect of different parameters (e.g., dimensions and locations of windows) on the air quality in the closed space. The models are also useful for knowing the accuracy of the measured CO$_2$ levels, which could be used for occupancy detection, based on the locations of the installed CO$_2$ sensors (Khalil, Wainer, and Dunnigan 2020). Another important application that has been addressed lately, is the use of CO$_2$ diffusion models as a way of determining the risk of viral infections in a closed space (Altamimi1 et al. 2020; Hatcher 2020).

With the need to adapt the CO_2 diffusion models to different closed spaces with different criteria, we have built a tool to automate and speed up the process of creating, changing, and simulating CO_2 dispersion models for any given closed space.

The Cell-DEVS CO_2 Diffusion (CD^2) tool we present here automates the creation and simulation cycle of CO_2 diffusion models. CD^2 offers, through a web interface, a cycle of features to the user that starts by converting a simple floorplan of a closed space into a Cell-DEVS model. Through a web interface, the user can specify the parameters of the model. The application server triggers the simulation which is performed on a remote server. Finally, the application server downloads the simulation results that the user can view. CD^2 does not require any technical knowledge from the user's side. It also does not require the user to install any of the M&S tools on their own workstation. It involves clear and simple steps to create the model, set parameters, and simulate the model in one web interface.

We start the discussion by laying the foundational background needed to understand the context of the tool, its applications, and related tools in section 2. In section 3, we present the software architecture and components, the workflow of the application, the technologies used, the CO_2 dispersion model, and a case study showing possible situations where CD^2 is an effective way of studying the CO_2 dispersion in a closed space. Finally, we summarize and conclude in section 4.

2 BACKGROUND

DEVS is a mathematical formalism that can be used to model complex systems. A DEVS model is defined by inputs, outputs, a set of states, an internal transition function, an external transition function, a time advance period, and an output function. The internal transition function changes the internal state of the model after the specified time advance has passed. The external transition function defines the model's reaction to inputs coming from other models that it interacts with. The output function is fired before changing the internal state of the model. DEVS structural models, which are atomic, can be combined in a hierarchical structure to form a more complex structural model. Those complex structural models are called coupled models. Cell-DEVS on the other hand is an extension of DEVS that combines DEVS with cell spaces (Wainer 2009).

Cell-DEVS has been used extensively to study many complex systems and modeling applications. This includes, but is not limited to, social (Khalil and Wainer 2020a), environmental (Wainer 2006), and biological applications (Goldstein and Wainer 2008). The applicability and advantages of Cell-DEVS as a formalism to model CO_2 diffusion in closed spaces have been presented in the literature. Among the advantages of Cell-DEVS is providing asynchronous execution, which results in a better execution time. Also, Cell-DEVS is supported by different tools and is proven to be applicable for modeling similar complex systems (Al-Habashna, Ruiz-Martin, and Wainer 2019; Khalil and Wainer 2020a). Hence, the authors have used Cell-DEVS formalism and its supporting tools to create and simulate several CO_2 diffusions models that are discussed in the next section (Khalil, Wainer, and Dunnigan 2020; Khalil and Wainer 2020b).

The set of Cell-DEVS models we refer to in this discussion aims at studying the spread of CO_2 in closed spaces. Such studies can be used to carry different modeling and useful simulation experiments. We detail here a few of the applications of CO_2 models and the associated experimental studies. First, it is essential to maintain a balance between indoor CO_2 level that is comfortable for occupants, while minimizing the energy consumed by ventilation systems. Second, CO_2 sensors are widely used for detecting occupants over other systems (because of low cost and non-intrusiveness), but their accuracy is very sensitive to the parameters of the space they are installed in. CO_2 dispersion models can help to define the best locations for the sensors and how they are expected to behave based on the parameters of the different indoor environments (Khalil, Wainer, and Dunnigan 2020). Finally, since occupants exhale CO_2 mixed with aerosols, the higher the concentration of CO_2, the more aerosols, including viral particles, occupants may inhale. As a result, CO_2 concentration has gained even more attention recently as it can be used as a proxy for the possibility of viral infection in closed spaces (Peng and Jimenez 2020).

CO_2 dispersion M&S models by Batog and Badura have been used to simulate the dispersion of CO_2 in a bedroom where one occupant is sleeping. The results of the study showed that small bedroom sizes result in unhealthy air quality in bedrooms, and thus extra ventilation is required. Also, the authors have reported that the location of the CO_2 sensor installed affects the validity of the readings logged by those sensors (Batog and Badura 2013).

In a previous study, we have tested the effect of sensor location using CO_2 dispersion models, using Cell-DEVS. The study has suggested that besides the effect of sensors' location, other parameters affect the CO_2 concentration levels in different areas of the room and affect readings of the CO_2 sensors (Khalil, Wainer, and Dunnigan 2020). Examples of such parameters are the presence of drywall dividers or open windows in the room. We have developed spatial indoor models for the effect of different parameters, such as the number of occupants and their seating arrangement, on CO_2 concentration dynamics. We have validated the model based on different laboratory/open office environments (Khalil and Wainer 2020b).

Other studies focus on CO_2 dispersion in classrooms, laboratories, and offices (Khalil and Wainer 2020b; Yalçın, Balta, and Özmen 2018). Yalçın, N., Balta, D., and Özmen have developed multi-compartment indoor air quality models based on the mass balance equation to simulate CO_2 concentration. The modeling considers different settings in the classroom environment. The authors also have tested the validity of their model and its capability of predicting CO_2 levels based on the given parameters of the indoor space (Yalçın, Balta, and Özmen 2018).

The model that we focus on here is a representation of a closed space with the following parameters: (1) room dimensions, (2) open windows locations and sizes in the room, (width, length, and height), (3) dimensions of open doors and their locations, (4) locations and dimensions of ventilation ports, and (5) the presence of occupants. We have calibrated the model in an earlier study using the settings and data collected from a real physical system of a computer laboratory at Carleton University. Then, we validated the model using another closed space in the same building during a different time and with different configurations. The analysis of the simulation results of the validated model has shown resemblance, with acceptable accuracy, with the physical system and has proved that the model is suitable for studying CO_2 spread in closed spaces (Khalil and Wainer 2020b). It is worth noting changing the model rules to incorporate other dispersions methods is independent of the CD^2 code. This model is built using the DEVS (Discrete-Event Systems specifications) and Cell-DEVS mathematical formalisms (Wainer 2009). These mathematical specifications are tool agnostic and independent of any M&S language. However, various tools have been developed to address the unique needs and uses of DEVS and Cell-DEVS.

The CO_2 models that we focus on in the development of CD^2 are built using CD++ (Wainer 2006); a toolkit that implements DEVS and Cell-DEVS theoretical concepts. In CD++, the rules that define a Cell-DEVS model are implemented using a scripting specification language that follows the Cell-DEVS formalism to define the model evolution (Wainer 2009). CD^2 uses a CD++ engine running remotely using the RESTful Interoperability Simulation Environment (RISE) middleware (Al-Zoubi and Wainer 2015). RISE exposes simulations as web services to allow those services to be interoperated with other communicating services. For example, it allows the remote execution of models over a distributed computing platform using Distributed CD++ (DCD++). Visualization tools allow the user to view the results of simulating the model by animating the changes in its states over the simulation time.

At present, there are no similar tools to serve the goals outlined earlier. However, other tools serve as containers for creating the whole lifecycle of cellular models, targeting users that have the expertise and want to design and create models rather than execute them. For instance, NetLogo offers a simple interface, in the form of buttons and sliders, to build models from rebuilt templates. However, it does not allow building hierarchical models. Besides, it does not support Cell-DEVS formalism (Sarjoughian, Sarkar, and Mayer 2010; Tisue and Wilensky 2004). As well, the Virtual Laboratory Environment (VLE) can define

DEVS models of different types, including Cell-DEVS. VLE can be used to define models in C++, and a graphical interface for model creation (Quesnel, Duboz, and Ramat 2009). The Component-Based System Modeler and Simulator (CosMos-CA) offers an interface to run the lifecycle process starting with model conceptualization and ending with the execution of the model for DEVS and Cell-DEVS models (Sarjoughian, Sarkar, and Mayer 2010).

CA Evaluation (CAEva) is designed for checking building conditions for fire accidents. The program is built using CD++. It allows drawing a board of cells where each cell can be of one of six types (e.g., fire or person on fire). The tool aims to help the modelers understand the consequences of not having sufficient escape routes in a closed space (Czerniak et al. 2016).

Evacuation Assessment and Real-Time Decision Guided Tool (EVACART) is a tool that focuses on emergency evacuation. EVACART is also built on top of CD++. Users of EVACART can create area maps. Then, they can select one of the evacuation strategies that are already available in the system. Through simulating different strategies, the user can evaluate the effectiveness of that strategy for the specific area map used. The simulation results can be viewed using the tool's 2-dimensional visualizer (Yang 2015).

The CD^2 tool, presented in the following sections, provides a simulation cycle for modeling a specific phenomenon: CO_2 dispersion in indoor environments. It uses a remote server for model execution. Besides, it requires minimal knowledge of software concepts since it abstracts all M&S technicalities from the end-users.

3 CELL-DEVS CO_2 DIFFUSION (CD^2)

As the CO_2 diffusion models have evolved and have become more useful in various applications, applying the models to different scenarios of a wide variety of indoor spaces has gained interest. In many experiments, it is very useful to vary the parameters in the model and run the simulations of the different scenarios to visualize the consequences of changing those parameters.

The process of applying CO_2 models to different indoor scenarios involves the following sequence:

1. Converting a floor plan of an indoor space (connected rooms or a single room) to a file with initial values for the state variables in each cell in the Cell-DEVS model. These values include all the specifications and locations of windows, doors, walls, ventilation ports, and workstations where occupants can sit.
2. Manually setting parameters that are not present in the image. For example, other information, such as whether there are occupants in the room or not and their number if they exist, is not part of the floor plan image. The user needs to edit the files composing the model to set parameters that define this information.
3. Connecting to RISE to run the simulation. Like other web services, to expose its services over the web, RISE requires a specific HyperText Transfer Protocol (HTTP) request as a contract, for communicating with other applications, in the form of a header and a body. The body is defined in Extensible Markable Language (XML). Creating this form of request is not for non-technical users that are not aware of the system's specifications. In addition, discovering errors created in such a process is time-consuming. CD^2 hides this detail from the user by creating the XML body of the HTTP request that acts as a contract between RISE and any other application. Then, CD^2 connects to RISE to run the simulation.
4. Downloading the simulation results from RISE to the user's local machine.
5. Visualizing the results of the simulation to be able to carry further analysis.

Handling these steps manually and trying to connect the output of each tool to the input of the tool used in the next stage is time-consuming, error-prone, and unintuitive. Our web application provides the user with a single interface giving them the ability to run the various steps automatically. The following steps, summarized in Figure 1, explain the workflow of CD2 in sequence:

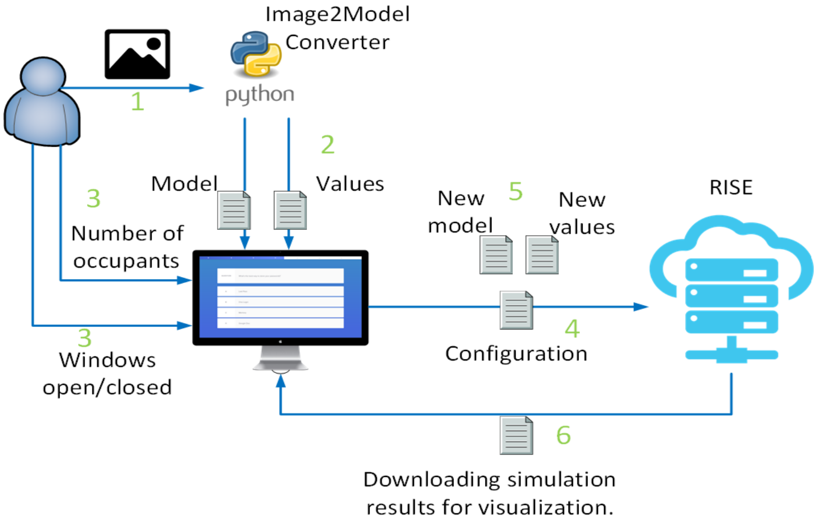
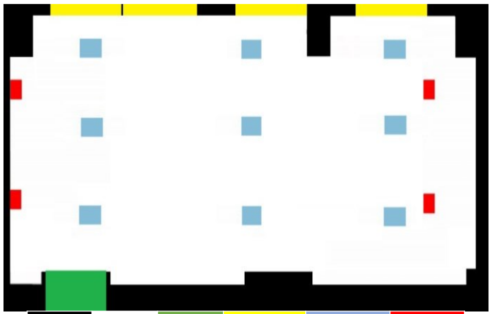


Figure 1: CD^2 workflow.

The first step in the workflow is to convert an image representing a floorplan to a Cell-DEVS model that CD++ can understand. The input of this step is a floorplan image of the indoor space to be modeled. The usual image file formats are accepted including JPEG, TIFF, and BMP. The output is a model file (.ma) and an initial values file (.val). The Cell-DEVS generator used considers images of floorplans with various colors representing air cells and obstacles (Henares, Cardenas, and Wainer 2020). The tool identifies the dimensions of the indoor space, the locations, and the dimensions of the obstacles. Then, it generates a model file and initial value files with this information. Six types of cells are generated, representing walls, air, doors, windows, diffusers ventilation ports, and grills ventilation.

Figure 2 shows an example of a room floorplan and the color-coding scheme used. When the user uploads the floorplan, they specify through the web interface the width of the indoor space in meters, and the conversion tool calculates the width, in the number of cells, of the resulting model. The length is calculated to scale based on the dimensions of the input image.



Black	White	Green	Yellow	Blue	Red
Walls	Air	Door	Window	diffusers	grills

Figure 2: Sample floorplans that CD^2 accepts as input and the legend used to create the model.

In the second step, the user can specify through the CD^2's web interface some settings of the indoor space being modeled; for instance, the status of the windows (closed or open; cells representing closed windows behave as solid impermeable structures, while open window cells are considered to have constant outdoor CO_2 concentration that is lower than the CO_2 concentration of the air cells indoors).

Also, the user can specify the number of occupants in the modeled space. Occupants are generated randomly in different empty locations in the occupied space. The tool inserts them as cells of type CO_2 source in the initial values file. The input to this step is the model created in step 1 represented by the model file and initial values file.

Now, CD^2 establishes communication with RISE. The input to this step is a configuration file that instructs RISE to build a session for the simulation and identifies the name of the model that will be simulated. At the end of this stage, a workspace for a new simulation environment is created on the RISE server. The configuration is passed as an XML file containing the description of the model, the simulation time, configuration parameters, and the names of the files defining the model.

At this step, RISE has already created a simulation environment to run the model. Then, the tool calls the appropriate APIs to submit the model files to the RISE web service. CD^2 sends all the model files to RISE, which runs the simulation. All the interactions with RISE are done without further user intervention.

When RISE finishes simulating the model, it stores the simulation results on the RISE server. Once the simulation results are available, the user would use CD^2 to download the results on their workstation. At this stage, the user would be able to visualize the results of the simulation.

3.1 Software Components

Figure 3 illustrates the different components of the CD^2 client-server software system. The client is a web interface that the users interact with through a browser to upload the floorplan images, specify the parameters, invoke the simulation, and download the simulation results from RISE. The application server is responsible for the core functionality of the system. It performs the conversion from image to model and changes the model files to reflect the parameters sent to it from the application's client.

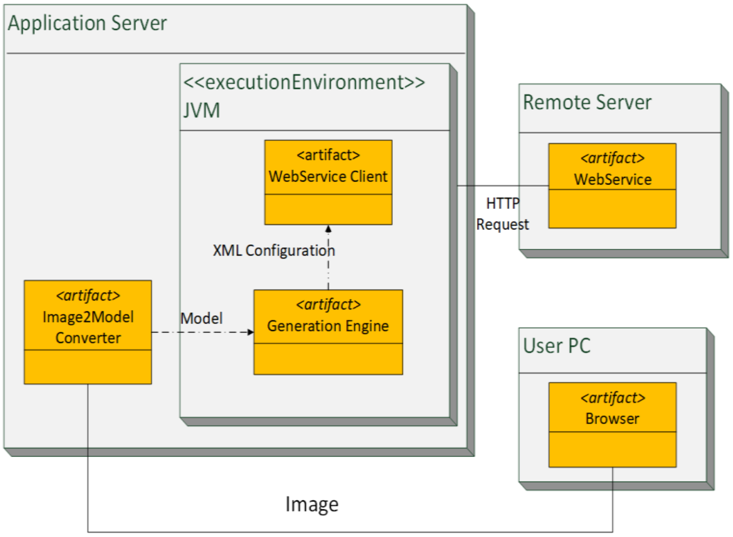


Figure 3: CD^2 software components.

The application server also prepares the model to be sent to RISE. It creates the HTTP requests, compresses the model files, and is responsible for all interactions with RISE. We have leveraged the servlet architecture. The client-side wraps all the information that the application server needs and sends it through an HTTP request. We use Java Server Pages (JSPs) for the webpages, while the server-side is coded in Java.

3.2 Case Study

In this section, we show an example of the process of creating, modifying, simulating, and getting the simulation results of a CO_2 model. Figure 4 shows a floorplan of indoor space with connected rooms. The indoor space has five windows and six ventilation ports.

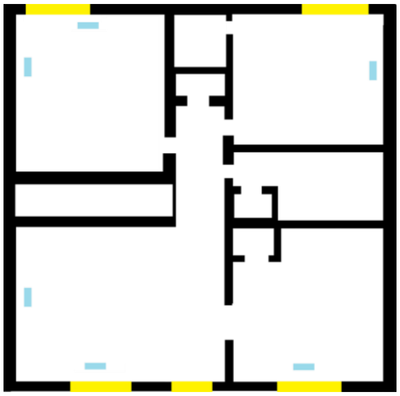


Figure 4: A floorplan of an apartment used as an example for the use of CD^2.

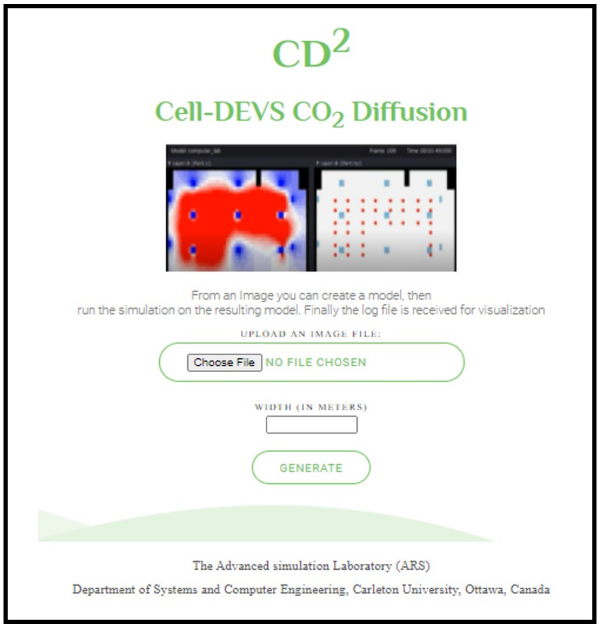
We show three different scenarios where CD^2 is useful to predict the effects of different parameters on the indoor air quality in this space building. The scenarios are a few samples of more possible scenarios that CD^2 can be used for. The ease of use and the abstraction of the details of the implementation of the M&S process make many other experimental scenarios possible through CD^2.

In the first scenario, we model the space in Figure 4 with five occupants, while the windows are closed throughout the simulation. Figure 5 shows a screenshot of the first webpage shown to the user, where they can choose the floorplan image and specify the width, of the real room that is represented by the floorplan.

The user enters the width of the space being modeled (in our example, 25 meters wide). The tool translates the input image into a grid of cells (in our example, a grid of 109×100 cells where each cell model represents a space of 25×25×25 cm^3). The grid size is determined based on the preset cell size (25×25×25 cm^3). In previous work, other cell sizes have been investigated and the current chosen size has proven to provide a good balance between the model's accuracy and the execution time. The tool automatically determines the grid size based on the dimensions of the input image and the preset cell size. Future versions of CD^2 can incorporate a feature to allow the user to specify the cell size.

After generating the grid (the model file and the initial values file), a second screen (presented in Figure 6) allows specifying information about the windows and the number of occupants. Based on the number of occupants entered by the user, the tool generates cells of type CO_2 source in different random locations on the grid. In our first scenario, the windows are closed, and there are five occupants. The viewed simulation results are based on the color palette defined in Figure 7 and Figure 8, representing the different levels of CO_2 concentration, in particles per million (ppm).

Figure 7 shows the simulation results of Scenario I where the windows are closed, and the number of occupants is 5 at time = 30, and at the end of the simulation. As observed from the figure, the area around the occupants has a high concentration of CO_2. The spot where an occupant is breathing has the highest concentration in dark red (dark red spots) due to constant breathing even while the remaining CO_2 diffuses in the room. The vents are working to offset this increase in CO_2 concentration as shown in the figure (blue areas).



CD^2

Cell-DEVS CO_2 Diffusion

From an Image you can create a model, then
run the simulation on the resulting model. Finally the log file is received for visualization

UPLOAD AN IMAGE FILE:

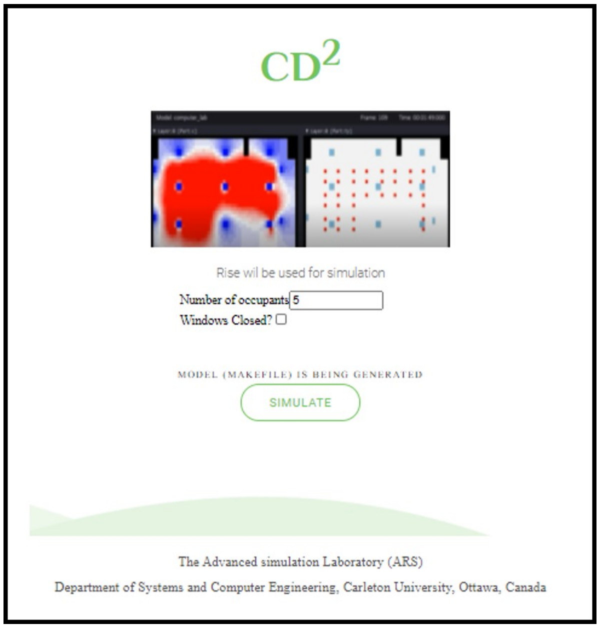
NO FILE CHOSEN

WIDTH (IN METERS)

The Advanced simulation Laboratory (ARS)

Department of Systems and Computer Engineering, Carleton University, Ottawa, Canada

Figure 5: CD^2 web interface where the user uploads the floorplan image and specifies the width.



CD^2

Rise wil be used for simulation

Number of occupants 5

Windows Closed? □

MODEL (MAKEFILE) IS BEING GENERATED

The Advanced simulation Laboratory (ARS)

Department of Systems and Computer Engineering, Carleton University, Ottawa, Canada

Figure 6: CD^2 web interface where the user sets the parameters of the model to be created.

In Scenario II, we keep the same parameters, except that windows are set to open. Figure 8 shows the simulation results. The simulation shows that open windows make a difference and introduce fresh air with less CO_2 concentration into the closed space. For example, the upper and lower right corners of the floorplan have lower CO_2 concentrations. From a user's point of view, simulating Scenario II requires the same process as simulating Scenario I while only unchecking the checkbox for windows status (Figure 6). The details of how this is done are hidden from the user. Also, applying this experiment to another floorplan requires only uploading a different image (e.g., in this example this would mean initializing more than 10000 cells).

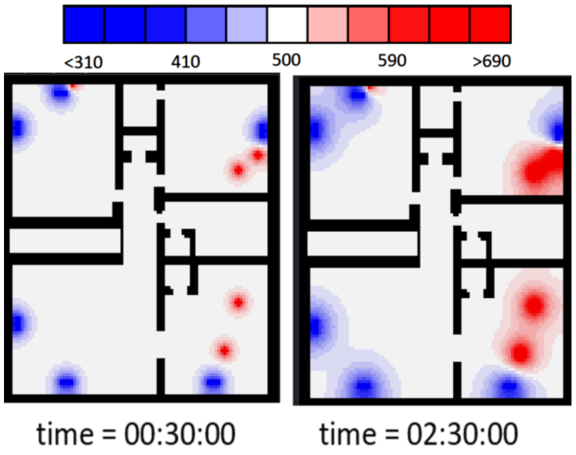


Figure 7: Simulation results of Scenario I: closed windows and five occupants.

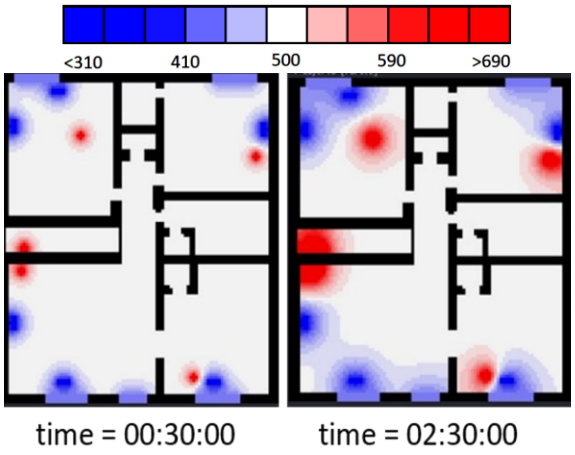


Figure 8: Simulation results of Scenario II: open windows and five occupants.

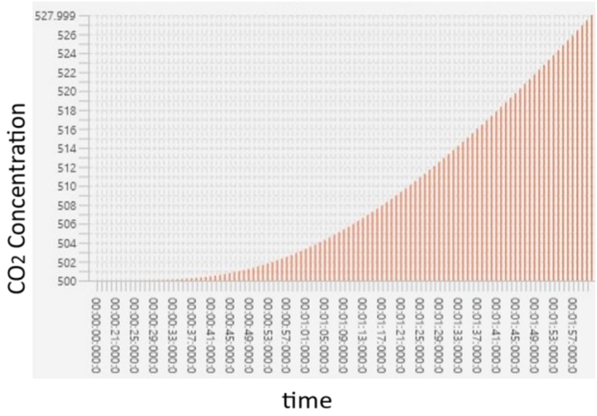


Figure 9: A plot of the CO_2 concentration level in ppm (y-axis) versus time (x-axis) for Scenario I (closed windows).

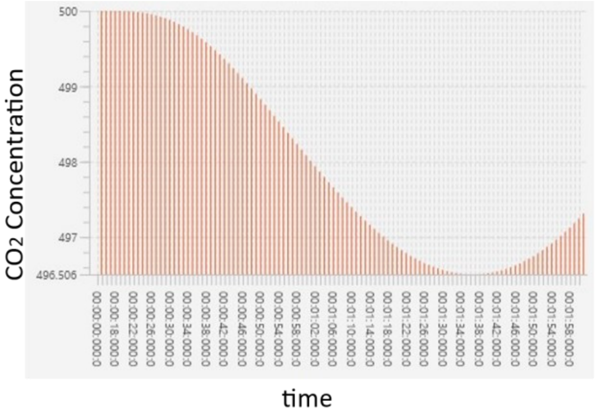


Figure 10: A plot of the CO2 concentration level in ppm (y-axis) versus time (x-axis) for Scenario II (open windows).

In Scenario I (see Figure 7), there are more occupants in the room and the window is closed. As apparent from the chart of Scenario I (see Figure 9), the CO_2 increase is offset by the existing ventilation only for a short period. Then, the CO_2 concentration keeps increasing even with the presence of the vents. On the contrary, in Scenario II, where we have only one occupant and the windows are open, the CO_2 concentration keeps decreasing, below 500 ppm providing better air quality and a healthier environment (see Figure 10). This decrease is because of the fresh air coming from the window. Then, closer to the end of the simulation the occupants' breath works against the effect of the fresh air coming from the windows. However, even then, the CO_2 concentration level does not reach the same level registered when the windows were closed. In Scenario III, we present another model generated by CD^2, where the user specifies the number of occupants to be 15 while the windows are closed. Figure 11 shows the simulation results of Scenario III. Experiments similar to this are useful in many applications such as determining the maximum acceptable number of occupants in a closed space. The tool allows for rapid change in the number of occupants. The simulation results may allow the user to conclude that the provided ventilation ports are not sufficient for this number of occupants for example. As shown by the simulation results in Figure 11, the CO_2 concentration levels are high due to the lack of ventilation and the high number of occupants. Demos of similar scenarios are available through our YouTube channel (ARSLab 2020).

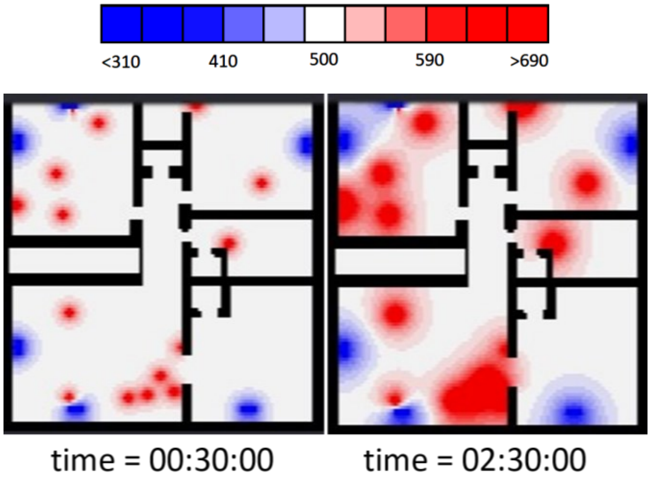


Figure 11: Simulation results of Scenario III: closed windows and fifteen occupants.

As we can see, CD^2 is useful for the specific application of creating CO_2 diffusion Cell-DEVS models. There are available robust tools that provide functionalities for the different stages in the cycle of model creation and simulation. However, those tools are created for technical users who have experience with creating models, through scripting, and the detailed rules used in the models.

At present, we are working on adding new functionalities that can be integrated into the application tool. One upcoming functionality is giving the user the option to upload a Building Information Model (BIM) as the base model instead of the floorplan image. At later stages, the tool can integrate with other computer-aided design tools. Figure 12 is the BIM, created using Autodesk Revit (Zotkin et al. 2016), of the example that we illustrated in the simulation in Figures 7, 8, and 11. Such models can be easily uploaded in the next version of the tool.

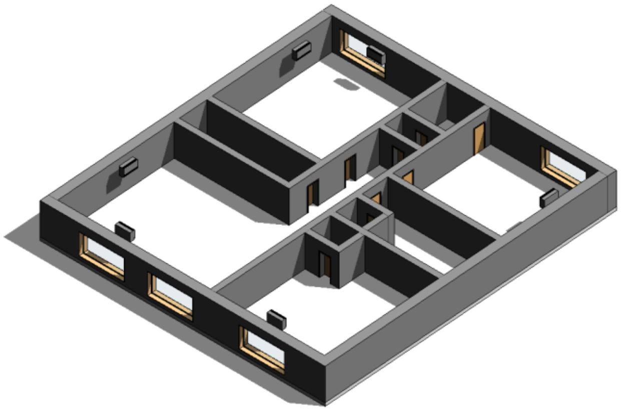


Figure 12: A BIM Model created using AutoDesk Revit© for the plan of Figure 4.

To be able to visualize the effect of open windows and the different locations where the occupants are, we have created a charting tool that can plot the change in CO_2 concentration level throughout the simulation in any given cell in the model grid (see Figure 9 and Figure 10). In the near future, we plan to integrate this charting tool into the CD^2 web interface. The charting tool reads the CO_2 concentration levels in the simulation results and plots it against the simulation time. We will also generate 3-dimensional models (which are supported by CD++ and RISE). The source code of the current version of CD^2 and the installation manual is available via GitHub (Khalil and Wainer 2020c). Also, future versions can allow changing the number of occupants and the status of the windows during simulation.

4 CONCLUSION

In previous studies, we have created and validated Cell-DEVS models for modeling and simulating the dispersion of CO_2 indoors. The created models have proven to be useful in many applications. Motivated by this and by the need for applying the CO_2 models to a variety of indoor spaces with different dimensions and settings, we have developed an application tool (CD^2) to automate the simulation cycle of CO_2 diffusion models. The application tool is specifically built to facilitate the process of applying different scenarios to our CO_2 models. Hence, the interface is simple and easy to use.

CD^2 allows its users to create a Cell-DEVS model from a simple 2-dimensional floorplan of any closed space. Then, it gives the users the option to specify the settings of that space (e.g., windows status, be it closed or opened, and the number of occupants). After the floorplan is uploaded and the parameters are set, CD^2 creates an HTTP request to send to the RISE server; including the XML configuration, based on the RISE specification, that is required in the body. Then, CD^2 runs the simulation on RISE. Finally, the user can download the simulation results to their local machine through the CD^2 web interface. Many possible future enhancements may be added to CD^2 including adding more options for changing the parameters of the modeled space through the web interface.

ACKNOWLEDGMENTS

The BIM diagram was made possible through the contribution of Nicolás Arellano Risopatrón and Stephen Fai, CIMS Lab. The authors are thankful for their support.

REFERENCES

Al-Habashna, A., C. Ruiz-Martin, and G. Wainer. 2019. "Analyzing the Impact of Quantum Size on the Accuracy and Performance of Cell-DEVS Fire Models". In *Proceedings of the 2019 Spring Simulation Conference*, edited by K. Lawsure, and M. Croll, pp. 1-12. Tucson, AZ, Institute of Electrical and Electronics Engineers, Inc. doi: 10.23919/SpringSim.2019.8732888.

Altamimi, T, H. Khalil, V. Rajus, R. Carriere, and G. Wainer. 2021. "Cell-DEVS Models with BIM Integration for Airborne Transmission of COVID-19 Indoors". In *Proceedings of the 2021 Symposium on Simulation for Architecture and Urban Design,* In press.

Al-Zoubi, K., and G. Wainer. 2015. "Distributed Simulation of DEVS and Cell-DEVS Using the RISE Middleware". *Simulation Modelling Practice and Theory* vol. 55, pp. 27-45. doi: 10.1016/j.simpat.2015.03.010.

ARSLab 2020. Advanced Realtime Simulation Laboratory YouTube Channel. https://www.youtube.com/user/ARSLab/videos. Accessed Apr. 17, 2021.

Batog, P., and M. Badura. 2013. "Dynamic of Changes in Carbon Dioxide Concentration in Bedrooms". *Procedia Engineering* vol. 57, pp. 175–182. doi: 10.1016/j.proeng.2013.04.025.

Czerniak, J. M., L. Apiecionek, H. Zarzycki, and D. Ewald. 2016. "Proposed CAEva Simulation Method for Evacuation of People from Buildings on Fire". *Advances in Intelligent Systems and Computing* vol. 401, pp. 315-326. doi: 978-3-319-26211-6_27.

Goldstein, R., and G. Wainer. 2008. "Modelling Tumor-Immune Systems with Cell-DEVS". In *Proceedings of the 2008 European Conference on Modelling and Simulation*, edited by L. S. Louca, Y. Chrysanthou, Z. Oplatková, and K. Al-Begain, pp. 329-335. Nicosia, Cyprus, European Council for Modeling and Simulation. doi:10.7148/2008-0329.

Henares, K., R. Cardenas, and G. Wainer. 2020. "Cell-DEVS Generator". https://github.com/Simulation-Everywhere/cell_devs_generator/commits?author=romancardenas. Accessed Nov. 21, 2020.

Hatcher, J. 2020. "Carbon Dioxide Monitoring to Lower the Coronavirus Threat". *Smart Buildings Magazine*. https://smartbuildingsmagazine.com/features/carbon-dioxide-monitoring-to-lower-the-coronavirus-threat. Accessed Nov. 2, 2020.

Khalil, H., and G. Wainer. 2020a. "Cell-DEVS for Social Phenomena Modelling". *IEEE Transactions on Computational Social Systems* vol. 7, pp. 725–740. doi: 10.1109/TCSS.2020.2982885.

Khalil, H., and G. Wainer. 2020b. "Modeling Carbon Dioxide Dispersion Indoors: A Cell-DEVS Experiment". In *Proceedings of the 14th International Conference on Cellular Automata for Research and Industry,* edited by T. M. Gwizdałła, L. Manzoni, G. Sirakoulis, S. Bandini, and K. Podlaski, pp. 226-236. Virtual, Springer International Publishing, AG. doi: 10.1007/978-3-030-69480-7_23.

Khalil, H., and G. Wainer. 2020c. "CD^2". https://github.com/hakhalil/SimulationChainTool. Accessed Nov. 21, 2020.

Khalil, H., G. Wainer, and Z. Dunnigan. 2020. "Cell-DEVS Models for CO_2 Sensors Locations in Closed Spaces". In *Proceedings of the 2020 Winter Simulation Conference*, edited by K. G. Bae, S. Lazarova-Molnar, Z. Zheng, B. Feng, and S. Kim, pp. 692-703. Virtual, Institute of Electrical and Electronics Engineers, Inc. doi: 10.1109/WSC48552.2020.9383937.

Peng, Z., and J. L. Jimenez. 2020. “Exhaled CO_2 as COVID-19 Infection Risk Proxy for Different Indoor Environments and Activities”. *Environmental Science & Technology Letters* vol. 8, pp. 391-397. doi: 10.1101/2020.09.09.20191676.

Quesnel, G., R. Duboz, and E. Ramat. 2009. “The Virtual Laboratory Environment – an Operational Framework for Multi-Modeling, Simulation and Analysis of Complex Dynamical Systems”. *Simulation Modelling Practice and Theory* vol. 17, pp. 641-653. doi: 10.1016/j.simpat.2008.11.003.

Sarjoughian, H., S. Sarkar, and G. Mayer. 2010. “A Novel Visual CA Modeling Approach and its Realization in Cosmos”. In *Proceedings of the 2010 Spring Simulation Conference,* pp. 1-4. Orlando, Florida, Society for Computer Simulation International. doi: 10.1145/1878537.1878790.

Tisue, S., and U. Wilensky. 2004. “NetLogo: Design and Implementation of a Multi-Agent Modeling Environment”. In *Proceedings of Agent 2004 Conference on Social Dynamics: Interactions, Reflexivity, and Emergence,* edited by C. M. Macal, D. Sallash, and M. J. North, pp. 7-9. Chicago, Illinois, North Americal Association for Computational Social and Organizational Science. doi: 10.1145/1878537.1878790.

Wainer, G. 2006. “Applying Cell-DEVS Methodology for Modeling the Environment”. *Simulation* vol. 82, pp. 635-660. doi: 10.1177/0037549706073698.

Wainer, G. 2009. *Discrete-event modeling and simulation: a practitioner's approach*. Boca Raton, CRC Press.

Yalçın, N., D. Balta, and A. Özmen. 2018. “A Modeling and Simulation Study about CO2 Amount with Web-Based Indoor Air Quality Monitoring”. Turkish Journal of Electrical Engineering and Computer Sciences vol. 26, pp. 1390-1402. doi: 10.3906/elk-1612-57.

Yang, N. 2015. *EVACART: Evacuation Assessment and Real-Time Decision Guided Tool*. Graduate research project, Embry-Riddle Aeronautical University, Florida, USA. Available via http://pages.erau.edu/~jafers/dokuwiki/lib/exe/fetch.php?media=yang_grp.pdf. Accessed May 16, 2021.

Zotkin, S., E. Ignatova, and I. Zotkina. 2016. “The Organization of Autodesk Revit Software Interaction with Applications for Structural Analysis”. *Procedia Engineering* vol. 153, pp. 915-919. doi: 10.1016/j.proeng.2016.08.225.

AUTHOR BIOGRAPHIES

HODA KHALIL is a postdoctoral fellow at the Advanced Real-time Simulation Lab and a contract instructor at the Department of Systems and Computer Engineering at Carleton University. Her e-mail address is hodakhalil@cmail.carleton.ca.

GABRIEL WAINER is a Professor at the Department of Systems and Computer Engineering at Carleton University. He is a Fellow of the Society of Modeling and Simulation International (SCS). His email address is gwainer@sce.carleton.ca.