



A cloud-based workflow for the integration of BIM to DEVS

Vinu Subashini Rajus^{*1}, Mitali Patel¹, Román Cárdenas², Omar Kawach³ and Gabriel Wainer¹

¹ Systems and computer engineering
Carleton University
Ottawa, Canada
vinusubashinirajus@cunet.carleton.ca,
on.ca,
mitalipatel3@gmail.carleton.ca,
gwainer@sce.carleton.ca

²Laboratorio de Sistemas Integrados
Technical University of Madrid
Madrid, Spain
r.cardenas@upm.es

³Omar Kawach
Faculty of Social Sciences
University of Victoria
Victoria, Canada
kawacho@uvic.ca

ABSTRACT

High-performance buildings and smart cities focus on reducing energy consumption while improving occupant comfort. An efficient building performance analysis would consider performance evaluation for the whole system than parts of the system. Advancement in technology has increased the need to integrate the energy and communication system for effective building performance. The integration of these systems is complex, and we propose using the Discrete Event Specification System (DEVS) to Building Information Modeling (BIM) for complex simulations. We describe a method to model these complex systems through DEVS and build a prototype implementation to integrate DEVS and BIM. The paper showcases two case studies highlighting BIM to DEVS integration and how it opens other design experiences through gamification or virtual reality.

Author Keywords

Building Information Model; Point Clouds; Autodesk Forge, DEVS

ACM Classification Keywords

I.6.1 SIMULATION AND MODELING (e.g., Model Development).

1 INTRODUCTION

Buildings are complex systems that include many interdependent physical parts and systems. For example, heat gain and heat loss are driven by the difference in indoor and outdoor temperature, thermal inertia, and heat capacity of the building's exterior envelope and interior mass. The complexity increases with flux in occupancy levels, lighting controls, use schedules, waste heat from appliances, and electronics [8]. Heat gain or loss affects how an occupant interacts with the building system like the thermostat or windows. For example, overheating can cause occupants to lower the thermostat or open windows. What action one may

take is unknown. All these factors are inter-related and play a role in understanding the system thoroughly.

To conduct a thorough system analysis as discussed above, it is useful to consider the building or urban system as a composite of interacting sub-models. The research [18] identify the importance of considering each sub-model at the design stage and suggests the use of Discrete Event Specification System (DEVS) to build a model of the whole system. DEVS [30] has been used to build models in different areas by domain experts in the field, ranging from evacuation models [28], fire spread [24], CO₂ diffusion [19], and COVID-19 viral spread models [3]. Although these kinds of models are beneficial for building design, the adoption by non-experts in the Architecture, Engineering, and Construction (AEC) industry is complex. If these models were available to designers, they could use them for evaluation at various design stages and performance analysis. Applying DEVS model to different building contexts rapidly could complement Building Information Models (BIM). A key consideration for DEVS to BIM interaction is the data exchanging format for BIM data extraction and visualization of the simulation results.

The research presented here introduces the design of an architecture and a prototype implementation that allows AEC experts to access, simulate, and visualize DEVS models without advanced expert knowledge while allowing access to BIM models for data extraction and visualization using the same platform. The prototype was created using Autodesk Forge [6], a cloud platform that gives access to BIM 360, a repository of BIM models. Cloud platforms provides services like storage, computing, networking, developer tools, identity, and security. The services vary based on the platform. The developers can focus on developing the web application and leave the logistics to cloud computing platform (like scalability, data storage or access) [21]. For instance, the Autodesk Forge cloud platform [6] enables the designers to access various API for integrating design and

communication among industry practitioners in the design and construction process. We can extract BIM data automatically and visualize the simulation on the same platform. The BIM data is parsed through a parser to convert it to a format compatible for Cadmium, a DEVS simulator used for this study. The DEVS model's repository is available for design exploration and the cloud platform provides a collaborative simulation approach for system performance analysis.

The rest of the paper discusses the current state of the art on BIM to DEVS integration, system design of the BIM to DEVS API, and BIM integration to DEVS and two case studies to show BIM data extraction and visualization. We further explain the design opportunities that BIM to DEVS integration provides through a COVID-19 serious game.

2 BACKGROUND

High-performance building and occupant comfort are becoming the focus leading to integrated building simulation. Advancement in technology has also furthered the assumptions that need to be taken into consideration for building simulation. For example, the energy and control systems are becoming more integrated and complex as it involves multiple domains like fluid dynamics, thermodynamics, control systems, communication systems, to mention a few [29]. These domains need experts and collaboration between these experts will amplify the design for the whole system [29]. Whether these models are designed by different experts or an individual user modeling the various requirements, needs a tool that gives flexibility in considering the various aspects. Occupant behavior in buildings is complicated to model, and they are one of the major contributors to energy consumption in the buildings. The Modeling and Simulation (M&S) community uses agent-based modeling or advanced simulation in considering the occupant behavior for building performance. There is a limitation in the current commonly used building performance simulation tools to integrate multiple models. High-performance buildings or energy-efficient buildings need to consider various factors for their performance, and these factors can be modeled better using a hierarchical and modular approach. Goldstein et al. articulate the importance of considering each sub-model and suggests a Discrete Event Specification System (DEVS) for the whole system analysis [18].

DEVS is a formalism that allows defining hierarchical and modular models, making them easy to reuse, adapt, and facilitate verification and validation. This definition makes it capable of representing a wide class of other dynamic systems, like differential equations or different types of automata. Its models are described as a composite of sub-models, each of them being behavioral (atomic) or structural (coupled). The atomic model's behavior is specified through a small set of states responding to external and internal events. The coupled models group other atomic and coupled models to create the hierarchy of sub-models [26]. DEVS has

been used to understand fire spread [24], evacuation [28], mold spread in buildings [1], and other complex systems. We developed a repository of various models like including evacuation [28], CO2 diffusion [19], and others. DEVS allows building complex models with ease using a hierarchical approach, that allows improved testing and reuse; as well, a discrete-event simulation allows running models efficiently. The models are built for interoperability and reuse, and hybrid models including continuous (HVAC, lighting) and discrete components (occupancy, switches, sensors) can be easily combined with each other. DEVS could also be used for collaborative building simulation among different expert users [15]. DEVS theory has been implemented in various simulation environments like PowerDEVS [10], DEVS-Suite [31], and CD++ [11] (See [16] for more details) and Cadmium [9].

Cadmium [9] is a strong-typed simulation engine based on the Discrete-Event System (DEVS) specification used in this paper. DEVS models are defined using two different building blocks: atomic models and coupled models. Atomic models describe the behavior of the system. Coupled models follow a structural model description pattern and specify how the system submodels (either atomic or coupled models) are interconnected. The DEVS formalism provides multiple advantages over other simulation formalism, such as modularity, reusability, and shorter model description times. Cadmium enables the definition of DEVS models, and of spatial simulation models based on the Cell-DEVS formalism [27], which allows the definition of Cellular models as DEVS specifications. It allows defining asynchronous cellular models that can be easily integrated with other pure DEVS models. In Cell-DEVS, the space is divided into a lattice of homogeneous cells. Each cell corresponds to a DEVS atomic model. On the other hand, the neighborhood of the cells (i.e., how cells are interconnected with each other) is defined with a DEVS coupled model.

Autodesk Revit has been used for accessing BIM models and use the BIM data to run simulations externally. Revit allows custom API integration to extract BIM data. [1] defined an add-in extension to Autodesk Revit that extracts the data based on selected IDS like walls or doors. Similarly, [Marzouk] built a 4D BIM model, an add-in extension to the Revit API for data extraction using C#. The simulation results are viewed using a different software tool for rendering; for instance, 3Ds Max, Maya, or BIM modeling tool like Revit. For instance, [28] used 3Ds Max: the BIM is imported into 3Ds max, and the simulation results are parsed through MaxScript for visualization. [14] used Dynamo [4], to integrate simulation results in Revit. The tools map the data to zones based on colors and track occupants' movement through a python script; they use lines and nodes represent the occupant's movement.

An ongoing research on making DEVS more usable to building industry is called DesignDEVS [16]. DesignDEVS is a simulation environment that integrates DEVS formalism

to promote understanding of DEVS principles and use [16, 17]. Currently, DesignDEVS is a standalone tool and is not integrated with any BIM software. The tool may enhance learning opportunities but is still limited to domain experts. DEVS models may help in early design decisions, building automation, and predictive analysis. The domain experts need a tool to extract BIM data and integrate simulation results on the BIM using the same application or platform.

BIM models are mostly available for DEVS simulator through Autodesk Revit. Their study is often limited to a single building context or the building models available to them. DEVS modelers are not BIM experts or designers. Hence, they need a platform to access BIM models easily. The building construction industry has advanced collaboration among various stakeholders through the cloud-based platform for BIM modeling and sharing. BIM cloud-based platform allows access to BIM models rapidly, store, and share the design with other stakeholders.

Autodesk Forge [6] is a cloud platform that allows access to a web-services based API to build a custom application. The web services API related we used in this research include the methods for Authentication, BIM 360, Model Derivative, and Viewer. The BIM 360 API allows users to access, upload, store, share 2D, and 3D BIM models for collaboration. The Model Derivative API translates the BIM models into SVF format for rendering and data extraction. The Forge Viewer [5] is a WebGL-based client-side JavaScript library for model rendering. The viewer includes some default settings but provides options for developers to add extensions. Figure 1 shows the Forge Viewer User Interface. Figure 1A shows the BIM 360 API to access the uploaded BIM models. Figure 1B is the viewer that shows the BIM model rendering and default setting of the viewer.



Figure 2. Forge Viewer user interface. (A) shows access to BIM 360 API, a repository for BIM models. (B) shows the rendering of the BIM model and default settings.

Our research aims to create methods to integrate data extraction and simulation, make the DEVS model's repository available for design exploration, and provide a collaborative simulation approach for system and building performance analysis. The BIM to DEVS API uses Autodesk Forge for BIM data extraction and visualization of the DEVS simulation results. The DEVS model were executed using the Cadmium simulator.

3 BIM-TO-DEVS INTEGRATION

This section describes the software architecture of our application that converts BIM to DEVS. We discuss different data extraction methods with their advantages and disadvantages, and data visualization.

3.1 Software Architecture

Figure 2 shows the two main components of the BIM to DEVS system architecture: the Autodesk forge API and the Cadmium simulator. The figure also shows the workflow to build and simulate models. As we can see in the figure, Revit models are uploaded to BIM 360 for extracting the data. We first select a BIM model for extraction from the BIM 360 API (1), after which we select a DEVS model (2) available in a model database (for instance, a model on building evacuation, CO2 diffusion, energy use, or room occupancy). Then, we extract the BIM data needed to execute the corresponding simulation (3); for instance, walls, windows, doors, or vents (section 3.2 explains this data extraction in detail). The data extracted is parsed through a parser (4) to convert the BIM data and integrate the DEVS scenario for the Cadmium simulator (see Section 3.3 for more details). Then, the DEVS Cadmium simulator runs the simulation (5), and the resulting output is generated for data visualization (6). The data is finally visualized using the Forge viewer (7) and integrated with the BIM (Section 3.4 explains the visualization methods in detail). Figure 3 shows the custom integration of the DEVS model in Forge viewer. 3A shows an existing DEVS model integrated, which simulates the diffusion of CO2 and the spread of COVID-19 viral particles.

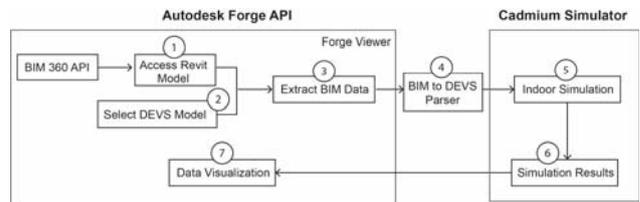


Figure 2. System architecture of BIM to DEVS application.

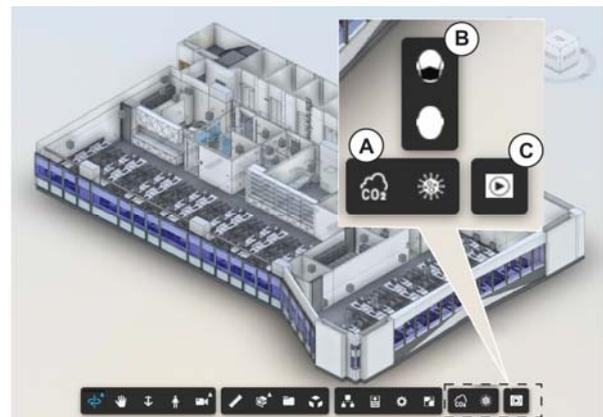


Figure 3. Custom interaction design for BIM to DEVS, 3A shows the DEVS model, 3B shows the subsection of DEVS model, 3C is the simulation play button.

3.2 Data extraction

There are different methods to extract data with Forge Viewer; in our case we use three different types of extraction: bounding boxes, ray intersection and box-collider.

Bounding Box

The bounding box is an invisible 3D rectangular space that contains the entire object and has all its sides facing in cardinal directions (+x, -x, +y, -y, +z, -z) [25]. This method returns a bounding box for each object by retrieving the object's database ID as a parameter. An intermediate floor plan is generated with those object's coordinates in voxel format. The BIM to DEVS parser uses the coordinates to create a Cadmium configuration file.

The advantage of the Bounding Box method is that the geometry from the BIM can be extracted very fast. One of the main disadvantages is that it cannot accurately extract curved objects, slanted objects, or objects not lined up with cardinal directions.

Ray Intersection

For Ray tracing, the first step is to determine the bounding box of the BIM objects and pass the ray, one from the top and another from the bottom. This will generate a ray for each pixel intersecting the bounding box of the BIM model. The ray intersection determines the objects inside the BIM, like walls, doors, vents, or windows for the extraction. The ray intersection determines the objects by selecting each object's database ID in an array and a ray, an instance of ray caster. Ray casting uses ray tracing to compute visibility [23]. This method is useful in extracting complex geometries. The extraction records each ray intersection as points to a voxel format with the object type. The extraction is parsed through the BIM to DEVS parser to create a Cadmium Cell-DEVS configuration file.

The ray intersection method extracts complex shapes accurately. The issue with this method is that it poses a problem in extracting data for more than one floor. It is resource-intensive and has an upper bound on how large of an architectural model it can handle. It rounds the position of objects at two steps, rather than just one, introducing truncation error.

Box-Collider

This method uses an SQLite database for extracting the metadata from the BIM geometry. We send queries from the client-side viewer with formal descriptions (the model's properties database to extract object ids), and the type of objects (walls, doors, windows, etc.). Their respective database ids are transmitted to the server for extracting the BIM data. The server then downloads the list of files related to the project and the properties database from Forge. Using the formal descriptions, we received as a parameter from the client-side, the server queries all the ids of each type from the properties database. For each type of object, the list of ids is broken up into chunks of up to 200, and each chunk is sent to the Forge Model Derivative API [7] to have a file containing only the geometry information generated (a OBJ

file). Once all these geometry files have been generated, download them one at a time and parse out the triangles list. Using the tribox3 [2] algorithm, each triangle calculates what cells it intersects and flags them as intersecting an object of that triangle's type (door, wall, window, etc.). Send the list of cells and flags back to the client.

Each cell type other than air is laid out in priority order (ex: Vents, Desks, Doors, Windows, Walls). This is taken to be of the type of the highest priority among types that it was flagged as touching for each cell. If it was not flagged as touching any cells, it is air (for example, a cell with parts of a wall and parts of a window in it is considered a window cell; a cell with a vent and a door and a window it is considered a Vent; a cell that is 99% empty air and has one corner clipped by a wall is considered as a wall cell).

Most of the work is done on the server-side. It is fast and can handle large models even with a small server. Since we need to handle large complex models (for instance, a University campus), we use the Box-collider method in future research. It will be built as an extension to the Forge Viewer, and non-experts can still use it without knowledge of SQLite.

3.3 BIM to DEVS parser

Once data has been extracted, we are able to use additional tools for creating simulation environments based on the original BIM model. As discussed earlier, in this research, we use the Cadmium simulation environment [9].

The parser creates a Cadmium Cell-DEVS configuration file according to the information provided by the BIM data extraction. This tool analyzes the ray trace of the BIM model and generates the corresponding configuration file for simulating the spread of CO₂ in the scenario. First, the script detects the dimension of the scenario according to the set of points extracted from the BIM. Then, it divides the space in cubes of 1 m³ and classifies each ray trace point with the cube that contains it. For each of these cubes, the script counts the number of points inside and classifies them according to their object type. Currently, the software can detect air, walls, windows, chairs, doors, and mechanical equipment.

By default, every cell is classified as air. However, if a cube contains three or more points of a different object type, this cube will be labelled as the object type of these points. In case a cube contains points of different types (e.g., wall and window), the object type with the higher number of points is selected. For instance, if a cube only contains one point of type wall, its corresponding cell will be air. On the other hand, if a cell contains 7 points of type Wall and 5 points of type window, the corresponding cell will be classified as a wall.

3.4 Data visualization

DEVS output varies based on the executed DEVS model. Cell-based output uses X, Y, Z coordinates for simulation results output (as seen in Figure 4). Spatial-based outputs use regions like rooms or zones for the results. As we need to

visualize Cell-DEVS models, we used grid or cell-based outputs.

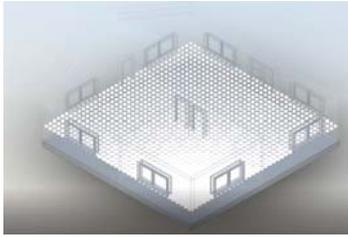


Figure 4. Cell-DEVS data mapped to the BIM 3D model

The simulation output is read, sorted, and filtered based on data type and visualization techniques like heatmap, point clouds, or particle animation. The filtered data will create an array composed of cell coordinates, time steps, and cell type. The cell types can vary based on the DEVS model type, and it will consist of building properties, air-cells, cell-state changes with values, and over time. To successfully create an animation, the cell coordinates in the array must have the proper offset to match the filtered data to the BIM geometry. Once the data is mapped, the data can be read and displayed in the Forge as a point cloud. The point clouds update at every time step as part of the animation. A countdown is provided in the UI to let viewers know the time remaining until the simulation has finished being played. Figure 5 shows the CO2 diffusion for a house model. Blue represents low concentration and red high concentration of the CO2 levels in the building. The height of the cell varies based on the CO2 concentration.

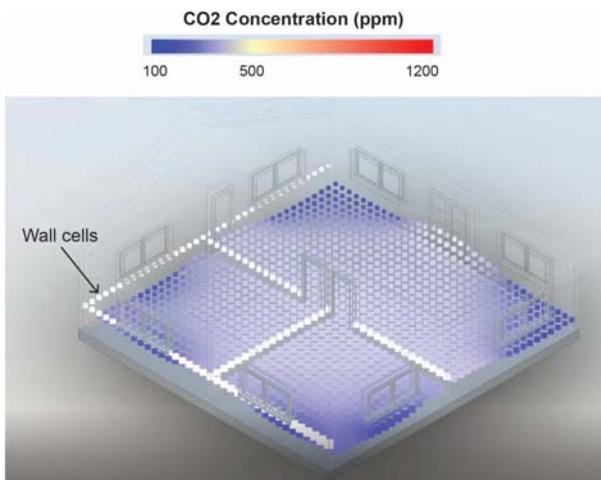


Figure 5. CO2 diffusion for a house with good ventilation, the color represents CO2 concentration, and the height is also manipulated based by CO2 levels

4 CASE STUDY MODELS

To show the use of our prototype implementation of the BIM to DEVS architecture and workflow, we built two case study models discussed in this section. The data extraction approach is similar for both models.

4.1 A CO2 Diffusion model

The idea of this DEVS model is to study CO2 diffusion in a room using the Cell-DEVS formalism, with the goal of finding the best placement of CO2 sensors for accurate occupancy detection [19]. The model shows the increase in CO2 concentration based on occupancy presence and ventilation system. The occupants are CO2 sources, which release CO2 with each breath (approximately every 5 seconds). CO2 will accumulate if there are fewer vents and more occupants. A well-ventilated building will reduce the CO2 spread. The model helps in analyzing the indoor air quality of the buildings. The simulation results in this example were visualized using Point Clouds in Autodesk Forge. We define a heatmap representing the CO2 concentration; a low CO2 concentration uses blue color, and higher concentration red. Figure 6A shows the results' overlaid on a rendered BIM model, and figure 6B shows the data in a wireframe.

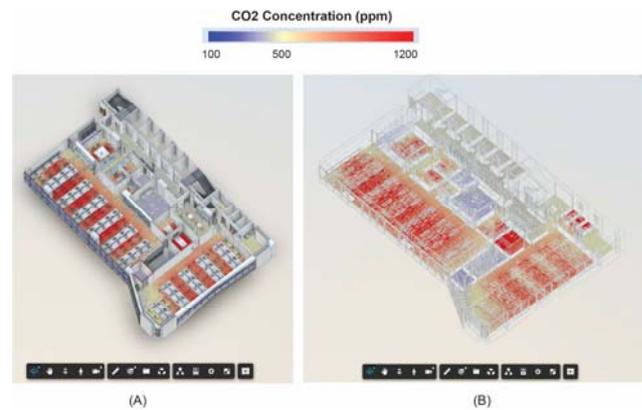


Figure 6. CO2 Diffusion spread in office building. 6A shows the simulation integrated on rendered BIM model and 6B on wireframe

The process and implementation of the CO2 diffusion model and simulation follows the workflow explained in Figure 2. A BIM model of an office in Sao Paulo, Brazil, was built and uploaded to the BIM 360 API. The BIM to DEVS API connects to BIM 360 to access the uploaded model (1), the user chooses the DEVS model (2), in this case, an existing CO2 diffusion model. The model extracts the following parameters to compute the diffusion process: the dimensions of the closed space, the location of windows, doors, walls, vents, and workstations (3). The parameters are defined as cell-IDs (position on XY plane) and space dimension as a shape for the Cadmium simulator. The BIM to DEVS parser (4) configures the BIM data with DEVS formalism for CO2 diffusion to a JSON format. The parsed file is inputted and run using Cadmium (5), the simulation output results (6) are read in the Forge viewer for data visualization (7). The visualization is integrated with the BIM model using Point Clouds. The simulation results consist of a timestamp, a cell position, and a cell-state showing the CO2 concentration level over time. All the steps are automated.

4.2 COVID-19 viral spread

This model is built to study the viral spread of the SARS-CoV-19 in buildings [3]. The model uses a small number of infected people in a room, combined with several healthy individuals. The simulation scenario presented in Figure 7 shows an execution of the model for an hour of simulated time. The infected people emit different amounts of viral particles every time they perform an action, including (breathing, coughing speaking). The ventilation system affects the direction and speed at which the virus spread indoors.

The COVID-19 viral spread model considers asymptomatic infected individuals shedding viral particles [12]. The model considers the number of infected people, breath rate, an infection threshold, the number of healthy individuals in the room, number, and position of vents available, and individual breathing behavior (coughing, singing, talking, mask ON, and mask OFF). Figure 7 shows the simulation results for a basic viral spread model. The model considers that the infected occupant is breathing with uniform air distribution. The COVID-19 viral spread was based on a real-world case reported in a restaurant in Guangzhou, China [20]. The BIM model in figure 7 was modeled based on the details provided in the article [20], which considered the spread of the viral particles in an actual building context where AC played a crucial role in spreading viral infection.

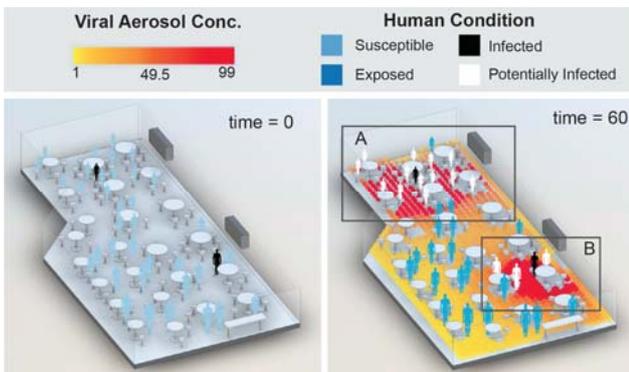


Figure 7. COVID-19 Viral spread integrated in BIM model

Here, we illustrate the process of BIM to DEVS integration for COVID-19 viral particles spread using the system architecture explained in Section 2, Figure 2. The restaurant model was built in Autodesk Revit and uploaded to the BIM 360 API. The restaurant model of Guangzhou, China is accessed through the BIM to DEVS API (1), the simulation model of COVID-19 viral spread is chosen for the DEVS model (2). The data extracted from the restaurant model are the dimensions of the closed space, the location of the vents, walls, tables, and chairs (3). The BIM to DEVS parser configures the BIM data with the DEVS model of COVID-19 Viral Spread to JSON format (4). The simulation is run using Cadmium (5), and the simulation results (6) are visualized in Forge viewer (7). The simulation results consist

of a timestamp, cell-coordinates, cell state changes: aerosol concentration levels, and the human condition.

The model uses two layers of data: the viral concentration in the room represented as a heat map using point clouds, and the individuals and their infective status. The concentration of viral particles is represented by a yellow to red palette, showing low to high viral concentration (Figure 7). The infective status of the individuals is based on inhaled aerosol concentration. The infected (index) cases are represented as individuals in black color. The susceptible individuals are colored in light blue (anyone present in the room is at risk), those exposed to the virus are in dark blue (representing anyone present in a cell with viral concentration). White is used for potentially infected individuals (the occupants have hit a viral concentration threshold). Figure 8A shows that the potentially infected person is higher in comparison to 8B. One of the reasons could be the space is compact and has more recirculated air.

4.3 Design Opportunities

The availability of DEVS models on a cloud platform for design opens other opportunities. In this section we discuss ideas related to serious gaming or VR for design exploration and enhancing building performance. The COVID-19 viral model presented in section 4.2. was extended as a game simulator to make a real-world risk assessment. It allows users to experience the implications by playing and exploring varying design parameters.

COVID-19 game simulator

The COVID-19 game simulator extends the COVID-19 viral spread model by adding gamification rules. The game assumes three infected persons at various locations for a large BIM model. The BIM model is an actual building at Carleton University campus. We use the workflow and tools introduced in Section 3 for BIM data extraction and visualization of simulation results. In addition to that we had to integrate game mechanics to the model. Collision detection using the ray casting method for navigation is used, as explained in section 3.2, to avoid the gamer passing through the walls or furniture in the BIM model. The game matches the 'camera' coordinates and the simulation results to calculate the intake of aerosols over time by the player. As, the user navigates, it shows how the intake of aerosols affect the gamer's health condition. The game's goal is to understand the implication of viral aerosol concentration in closed spaces. The player's goal is to perform the tasks and maintain their health within the time count.

The model considers different ventilation rates spatially to observe its implication on the spread of the virus. The ventilation location, speed, and orientation affect the rate and direction at which the virus spreads. The simulation results were large, and we split them into chunks for streaming, which increased the performance of visualizing and streaming large data. Figure 8 shows the gaming user interface in Autodesk Forge API. Figure 8A and Figure 9 shows the game overview. The overview displays the

gamification results: a “health value” bar, a taskbar, and the time left to complete the game. A mini map was integrated for navigation (figure 8B). We show the viral concentration level (8C): yellow for less concentration, and red for high concentration.

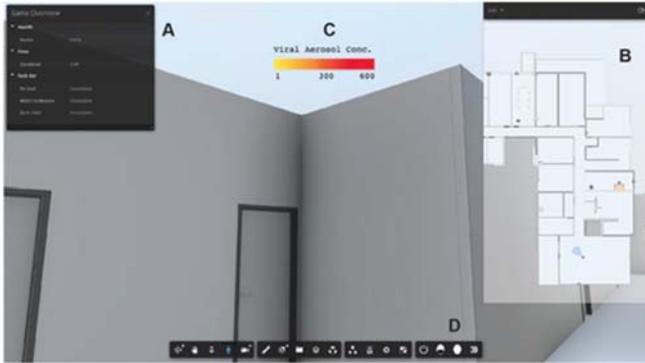


Figure 8. User interface for COVID-19 Game Simulator

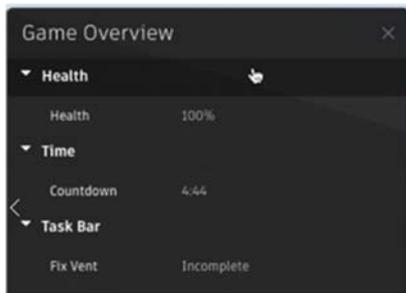


Figure 9. Game overview task pane

Figure 10 shows the difference in viral concentration when the infected individuals are wearing masks (an option in the game). Figure 11 shows a 3D perspective of the simulation results integrated into the BIM.

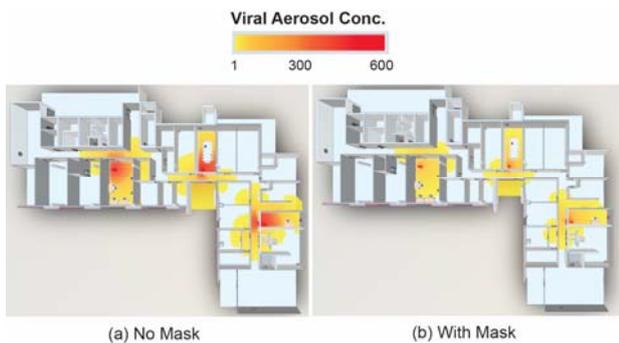


Figure 10. The two game modes: Mask OFF and Mask ON

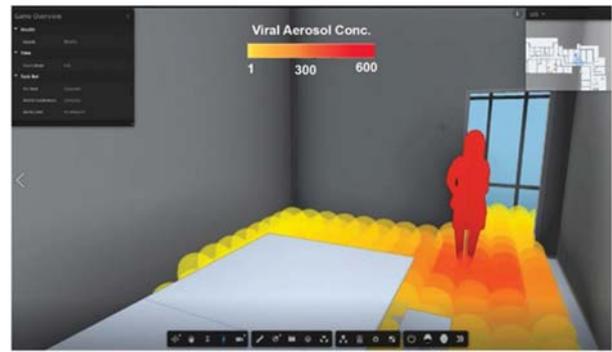


Figure 11. 3D view screenshot of the game interface with simulation results

5 DISCUSSION

The case studies above show how to integrate DEVS models into BIM software. The proposed approach gives practitioners easy access to complex simulation models to evaluate their design and use it for building automation and design exploration. The COVID-19 game simulator is a good illustration of how to train facility managers in the exploration of HVAC systems and spatial lockdowns to assess the virus's spread in the buildings. The ongoing pandemic imposed physical distancing rules in the buildings, and architects used parametric modeling for the best adaptable interior layout. The exploration was purely design-oriented, but when combined with a real-world model like the COVID-19 viral spread, they can make a more realistic risk assessment and even use the results for training.

Integrating the model on a cloud platform increases access to the repository of BIM models and interaction among various design experts. The DEVS models can be applied to various building contexts rapidly and analyze how contextual design affects the simulation results. Another advantage of DEVS integration with cloud platform is different experts like lighting designers, HVAC system, or building automation can collaborate on a design. Since DEVS couple different models together, different models by the experts can be integrated for a collaborative building simulation [15].

Autodesk Forge uses THREE.js version 71, and we had difficulties in implementing newer elements from the latest version. Currently, Autodesk Forge provides the extension Hyperion for data visualization, increasing the visualization capabilities, and maybe a solution for the old version of THREE.js. Modeling and simulating large BIM data produces large simulation result outputs that may affect the browser's performance. The issue can be resolved by splitting the large data into smaller chunks or increasing the time-step of the simulation to reduce the file size.

The BIM to DEVS prototype currently serves as a model to integrate BIM with complex simulations. The data is manually inputted to the cadmium simulator, and the results are manually loaded in the Forge viewer. The example demonstrated here is for small and large buildings. The next step would be to apply for a large context design like campus

or cities, but that will require considering the extraction method of BIM in regard to scale and depth.

6 CONCLUSION

Integrating the DEVS model benefits architects, engineers, and designers for design and building performance evaluation. The complex simulation models are available for them to make design exploration with real-time simulation. The integration makes the DEVS model easily accessible and applicable for design analysis and to various expert users. The software architecture and workflow presented here allow integrating complex simulations on a cloud platform. Ray Intersection can be used on the client-side for small BIM models and large models using a box-collider method. The simulation results vary based on the DEVS model.

A prototype of the architecture was built using the Autodesk Forge cloud platform. The prototype has a custom-designed interaction for DEVS model. The three case studies presented show how to define BIM to DEVS for design and performance analysis. The data extraction approach varies based on the building context. The data visualization of the simulation varies based on the DEVS model. For instance,

ACKNOWLEDGMENTS

We would like to acknowledge Ryan Carriere for COVID-19 DEVS model, Nicolás Arellano Risopatron and his colleagues at CIMS Lab, Carleton University, as well as Tiago Ricotta for the BIM models. Griffin Barrett for BIM data extraction. This research has been partially funded by NSERC and Autodesk Research.

REFERENCES

1. Ahmed, A. S., Wainer, G., and Mahmoud, S. 2010. Integrating building information modeling & cell-DEVS simulation. Proceedings of the 2010 Spring Simulation Multiconference (2010), 1 - 8.
2. Akenine-Möllser, T. Fast 3D triangle-box overlap testing. *Journal of graphic tools*, 6(1), (2001), 29-33.
3. Advanced Real-Time Simulation Laboratory. https://github.com/SimulationEverywhere-Models/indoor_virus_spread.git. Retrieved March 2, 2021.
4. Autodesk Inc. Open source graphical design for programming. <https://dynamobim.org/>
5. Autodesk Inc. Autodesk Forge Viewer. https://forge.autodesk.com/en/docs/viewer/v7/developers_guide/overview/
6. Autodesk Inc. Autodesk Forge. <https://forge.autodesk.com/developer/documentation>
7. Autodesk Inc. Autodesk Model Derivative API. <https://forge.autodesk.com/en/docs/model-derivative/v2/reference/http/job-POST/>
8. Bachman, L. R. *Two Spheres: physical and strategic design in Architecture*. Routledge, 2012.
9. Belloli L., Vicino, D., Ruiz-Martin, C., and Wainer, G. Building devs models with the cadmium tool. In 2019 Winter Simulation Conference (WSC), (2019), pp. 45-59. IEEE
10. Bergero, F., and Kofman, E. PowerDEVS: a tool for hybrid system modeling and real-time simulation. *Simulation* 87 (1-2). 2011, 113-132.
11. Bonaventura, M., Wainer, G.A., and Castro, R. Graphical modeling and simulation of discrete-event systems with CD++ Builder." *Simulation* 89, 1 (2013), 4-27.
12. Bromage, E. The Risks-Know Them-Avoid Them. Erin Bromage: COVID-19 Musings blog. 2020
13. Chopard, B., and Droz, M. *Cellular automata*. Berlin, Germany, Springer (1998).
14. Eftekhariad, R. Improving Fire Emergency Management Using Occupant Information and BIM-Based Simulation." PhD dissertation, Concordia University Montreal, Quebec, Canada. 2019.
15. Goldstein, R., and Khan, A. Introducing DEVS for collaborative building simulation development. SIMAUD, Orlando, (Tutorial) 2010.
16. Goldstein, R., Breslav, S., and Khan, A. DesignDEVS: Reinforcing theoretical principles in a practical and lightweight simulation environment. In 2016 Symposium on Theory of Modeling and Simulation (TMS-DEVS) 2016, pp. 1-8. IEEE.

our CO2 diffusion implements only a heat map whereas the COVID-19 viral spread implements heat maps as well as human conditions.

The proposed method allows to integrate any other simulation software to the cloud platform. The integration of BIM to DEVS opens other design opportunities like gamification and virtual reality. The building designers can use these methods for simulating real-time experience by modifying the designs. The real-time incorporation of BIM data with DEVS in VR will enhance collaborative decision making. The cloud platform and DEVS increase collaborative building simulation and accessing various building contextual models. The Forge platform provides access to various API services and hence increases the interoperability of the tool.

In the future, the BIM-based cloud-platform could be used for building a digital twin of the real-world model. Real-time simulation with historical and contextual data will enhance predictive analysis. Integrating other DEVS models will be considered.

17. Goldstein, R., Breslav, S., and Khan, A. 2018. Practical aspects of the DesignDEVS simulation environment. *Simulation* 94, no. 4. pp 301-326.
18. Goldstein, R., Wainer, G. A., and Khan, A. The DEVS formalism. In *Formal Languages for Computer Simulation: Transdisciplinary Models and Applications*, IGI Global. 2014, 62-102.
19. Khalil, H., and Wainer, G. Modeling Carbon Dioxide Dispersion Indoors: A Cell-DEVS Experiments, In *Proceedings of the Cellular automata for research and industry ACRI, 2020*. https://doi.org/10.1007/978-3-030-69480-7_23
20. Lu, J., Gu, J., Li, K., Xu, C., Su, W., Lai, Z, Zhou, D., Yu, C., Xu, B., and Yang, Z. COVID-19 Outbreak Associated with Air Conditioning in Restaurant, Guangzhou, China, 2020. *Emerging Infectious Diseases*, 26 (7), 1628-1631. <https://dx.doi.org/10.3201/eid2607.200764>
21. Lucovsky, M., Collison, D., Spivak, V., Chen, G. C., and Laddad, R. Cloud platform architecture. (Jan. 2014). Patent No. US 8,627,426 B2. Filed April 26th 2010, Issued Jan. 7th 2014.
22. Marzouk, M., and Al Daour. 2018. Planning labor evacuation for construction sites using BIM and agent-based simulation, *Safety Science* 109, 2018,174-185. ISSN 0925-7535, <https://doi.org/10.1016/j.ssci.2018.04.023>.
23. Weghorst, H., Hooper, G., and Greenberg, D. P. Improved computational methods for ray tracing. *ACM Transactions on Graphics (TOG)*, 3 (1), (1984), 52-69.
24. Sun, Q., and Turkan, Y. A BIM Based Simulation Framework for Fire Evacuation Planning. In *Advances in Informatics and Computing in Civil and Construction Engineering*, Springer International Publishing, Cham, 2019, pp. 431-438.
25. Tammik, J. 2016. Solid From Bounding Box and Forge Webinar 4. (September 2016). Retrieved July 6, 2020 from <https://thebuildingcoder.typepad.com/blog/2016/09/solid-from-bounding-box-and-forge-webinar-4.html>
26. Wainer, G. *Discrete-event modeling and simulation: a practitioner's approach*. 1st ed., CRC Press, Inc. 2009.
27. Wainer, G. A. An Introduction to Cellular Automata Models With Cell-DEVS. In *2019 Winter Simulation Conference (WSC)*, National Harbor, MD, USA, (2019), pp.1534–1548. <https://doi.org/10.1109/WSC40007.2019.9004763>
28. Wang, S., Van Schyndel, M., Wainer, G., Rajus, V.S., and Woodbury, R. DEVS-based building information modeling and simulation for emergency evacuation. In *Proceedings of the 2012 Winter Simulation Conference (WSC)*. IEEE, 2012, pp. 1–12.
29. Wetter, M. A view on future building system modeling and simulation. Technical Report. Lawrence Berkeley National Lab (LBNL), Berkeley, CA (United States). 2011.
30. Zengin, A., and Sarjoughian, H. DEVS-Suite simulator: A tool teaching network protocols. In *Proceedings of the 2010 Winter Simulation Conference (WCS)* (2010). 2947–2957. <https://doi.org/10.1109/WSC.2010.5678989>