SOFTWARE ARCHITECTURE FOR INTEGRATING DEVS SIMULATION INTO BIM

Mitali Patel
Vinu Subashini Rajus
Gabriel Wainer

Department of Systems and Computer Engineering
Carleton University
Ottawa, ON, CANADA
mitalipatel3@cmail.carleton.ca,vinusubashinirajus@cunet.carleton.ca,gwainer@sce.carleton.ca

ABSTRACT

Building performance is just as important as designing and constructing buildings that are aesthetically appealing. Modern buildings should be more energy efficient in terms of long-term sustainability and energy use. A high-performance building requires proper design and construction, with more energy-efficient HVAC systems, while keeping human comfort in mind. Reconstruction will be difficult and expensive if the performance measures are not achieved after the building is constructed. As a result, evaluating a building's performance in the pre-design phase and simulating the building before construction can be beneficial. We present a solution that combines the Discrete Event System Specification (DEVS) with Building Information Modeling (BIM) to predict building efficiency. We present two case studies using simulations to show how the proposed approach helps in enhancing the building's performance.

Keywords: DEVS, Cell-DEVS, Building Information Modeling (BIM), Autodesk forge, Data visualization.

1 INTRODUCTION

Building design and energy performance evaluations are critical for lowering energy consumption and making buildings more sustainable. For instance, in cold countries, many building design parameters including architectural design, heating, occupancy hours, occupant comfort preference, number of tenants, operational behavior (opening of windows, doors, heating, and AC) and, others, play a role in energy use. Likewise, factors like hours of occupancy, activity, movement, windows and door opening behavior and vents play a major role in the use of energy and the emission of CO2 (Khalil and Wainer 2021).

Modeling and Simulation (M&S) can conduct a thorough analysis of the energy usage in the buildings, by considering the factors mentioned above. Most of the building energy is used to maintain a comfortable indoor environment in terms of thermal comfort (heating or cooling) and air quality (ventilation). As a result, it is crucial to adopt energy-efficient solutions without sacrificing the quality of the internal environment while designing a building.

To design a building energy model, it is essential to analyze the elements that determine the energy building usage including climate, building related features (location of walls, windows or doors), building systems (heating, lighting or mechanical equipment), and occupant related characteristics. Because different energy sources have varying consumption characteristics and environmental impacts, they must be properly evaluated while modeling. After the model has been developed, an energy simulation can be run to examine

ANNSIM '22, July 18-20, 2022, San Diego, CA, USA; ©2022 Society for Modeling & Simulation International (SCS))

the behavior of the building. The results can be compared with the actual building performance. To study the different types of building scenarios, a modeler can change the building components, simulate each instance to acquire the energy consumption, and compare the results. This allows for the optimization of performance evaluation and decision-making.

Simulating multiple scenarios with ease allows designers and architects to determine, which version of an energy model would be the best for achieving the desired goals. This prediction could help to identify which design strategies would reduce energy use while remaining cost-effective. This is a key aspect of high-performance building design, as well as a way to enhance design decision-making. The overall process of modelling and simulating building energy would benefit the pre-design stage of construction. It assists in the analysis and evaluation of building designs through a number of alternative simulations in an iterative process (Jiang et al. 2012). To better estimate the performance, each stage can be divided into smaller sections (for example, pre-design, main design, and detail design).

Discrete Event System Specification (DEVS) framework in M&S (Zeigler, Prähofer, and Kim 2000) is a good candidate for modelling and simulation of building design. DEVS allows the system to be described in a formal, mathematical, structured way. Subjectivity in the model construction is minimized in this manner. It maintains a system's hierarchical and modular view; that is, it allows each of the system's components and the multilayer architecture, to be modelled separately. Additionally, several sub-models may be developed from various domains, such as occupant models, HVAC models, and building models, all of which are needed for energy simulations. Designers may test each sub-model, evaluate their solutions, and enhance them by changing the inputs and outputs at the end of each simulation stage.

DEVS (or other modelling and simulation techniques) requires precise information on the building elements, occupancy management data, the number of mechanical equipment, and the building dimensions to assess the building’s performance accurately. Building Information Model (BIM) software can be used to obtain all of the building's data. BIM is a three-dimensional digital representation of a building (Azhar 2011), a virtual environment in which the building can be replicated. BIM models contain information on geometry, spatial positions, geographic information, building elements, cost estimates, material inventories, and project scheduling for a building under design. Combining DEVS and BIM may help designers and architects create energy-efficient and sustainable buildings.

In this work, we leverage the mentioned above data obtained from the BIM of the real building to study the building’s performance using the DEVS methodology. We are also interested in studying the interoperability of DEVS and BIM because most complex building systems can be specified using these methodologies. This type of interoperability provides a platform for performance simulation as well as quick and accurate visualization of simulation results in the BIM model for contextual analysis.

We want to provide a robust platform for integrating and visualizing DEVS models into BIM using a single platform that can adapt to varying DEVS models. We use CD++ and Cadmium as case studies. The models could be used easily applied to a variety of architectural contexts, making them a useful addition to BIM.

The accessibility of DEVS models to non-modeling experts (such as architects and designers) might be beneficial for their design activities. They can leverage the use of these models into BIM for evaluation at various design stages and performance analysis. This will benefit them in the pre-design phase of a structure in a building.

We present a method to integrate BIM with the models built using DEVS, defined as a software architecture design and workflow, and we demonstrate how it can be used to integrate several DEVS models into BIM. The simulation outcomes of different models can be easily incorporated into the architecture and visualized with BIM. This tool allows designers, modelers or architects to access, simulate and visualize DEVS models on the same platform while allowing access to BIM models without any expert knowledge.

We built a tool using Autodesk Forge (Autodesk Inc. 2022), a cloud-based development platform that allows users to access or upload BIM models efficiently using token-based authorization and authentication.

It provides a set of web-service APIs that users can be used to build custom applications using their data. We used Autodesk Forge Viewer (Autodesk Inc. 2022), a client-side JavaScript library, that renders the 3D model in the forge viewer once the Revit model is translated to SVF format using the server-side Autodesk Forge Model Derivative (Autodesk Inc. 2022). Autodesk Forge also allows designers to access various APIs for integrating their building designs and collaborate with industry professionals during the construction process. We illustrate how to extract the building information from BIM automatically and parse the results into a format suitable for DEVS simulators. Similarly, we show how to visualize simulation outcomes by converting them back into BIM, which may be used for scenario validation and analysis. We represent the simulation results through point clouds into BIM for better visualization. Both the process data extraction of BIM and data visualization are done on a single platform.

We show the use of the platform and architecture with case studies that show how the proposed architecture adapts to the different DEVS models. One is a CO2 diffusion model simulated using CD++ and the second one is a Viral Spread Model for COVID-19 indoors simulated using Cadmium. The results emphasize its enhanced reusability and extensibility by providing better simulation results, allowing designers to assess building performance and compare solutions.

In Section 2 of this research, the background of DEVS Formalism, BIM, and work related to DEVS and BIM is explained. Section 3 discusses the software architecture for combining the DEVS and BIM. In section 4, the design is implemented using the Cell-DEVS model in a variety of BIM scenarios. Finally, in section 5, we present our conclusions and suggestions for future improvements.

2 BACKGROUND AND RELATED WORK

M&S can be described as the process of abstracting the physical model's description to analyze the system's dynamic behavior and performance to be applied in the real world (Zeigler, Prähofer, and Kim 2000). A model is a conceptual presentation of a system of its dynamic behavior. The model’s behavior is generated using a set of rules or mathematical equations (Wainer 2009). Simulation is a process that drives the execution of these models by providing a set of inputs. A simulation can be better understood through visualization, a process to show the different simulation results that help analyze a complex system's behavior.

Theoretical modeling and simulation formalisms are well suited to design and evaluate complex systems, defined as a collection of interacting elements and consist of emerging global dynamics (like ant colonies (Boccara 2010) or occupant behavior in buildings). In this work, we use the DEVS theory of Modeling and Simulation for analyzing complex systems.

2.1 DEVS Formalism

The DEVS introduced by Bernard P. Zeigler is a mathematical formalism to build dynamic real-life systems efficiently based on modeling and simulation theory (Zeigler, Prähofer, and Kim 2000). The models developed using DEVS (Concepcion and Zeigler 1988) are based on a discrete event approach where time is continuous. DEVS allows many submodels to be linked together to build a hierarchy. Each model can be behavioural (atomic) or structural (coupled), with a time basis, inputs, states, and outputs, as well as functions to calculate the next states and outputs. The models can be linked in a hierarchical manner, allowing for model reuse (Wainer 2009).

Cell-DEVS is an extension to DEVS that allows the implementation of cellular models with explicit timing delays (Wainer 2009). It is used to build models based on discrete event cell spaces. Cell-DEVS aims to make it easy to develop cellular models and increase their execution speed and precision by using a discrete event method. Each cell in a grid space behaves as an atomic model having independent behavior and it can be later integrated into a coupled model representing the cell space (Wainer and Fernández 2016).

Several DEVS simulators can execute the model's instructions to generate the behavior of the modelled complex systems. DEVS simulators are based on abstract simulation algorithms (Wainer 2009). Examples of these simulators are DEVS-Suite (Kim, Sarjoughian, and Elamvazhuthi 2009), CoSMos (Sarjoughian and Elamvazhuthi 2009), CD++ (Chidisiuc and Wainer 2007), PowerDEVS (Bergero and Kofman 2011) and Cadmium (Belloli et al. 2019). We have used CD++ and Cadmium Simulators in our research that are implemented based on the specification of DEVS, and Cell-DEVS. CD++ is a rule-based simulation engine to develop and execute DEVS models, built as a class hierarchy (Wainer 2002). This modelling tool is composed of a number of separate software components that run on different operating systems. The CD++ tool executes in a stand-alone mode with an independent graphical interface, and allows users to debug, execute and visualize the results remotely. It is implemented using DEVS formal modeling technique to improve the development time of the simulations. In this simulator, DEVS models can be defined in a custom language. The model behavior is defined in a model file that contains simulation configuration parameters with a set of rules. Cadmium can define more complex model using C++. It is a C++17 header-only library, which makes it easy to include and integrate other C++ projects which require simulation with DEVS (Belloli et al. 2019). Users can define complex message data types with different ports and non-uniform cell space topologies. Both the behavior and the initial values of the system are defined in a JSON file.

These simulators require two main inputs for simulation: a model (which generates behavior) and a base scenario (in a spatial model, the space under study, like a city or a building). Each scenario is defined as a space-discrete representation of a building under study. For example, to simulate behavior in a building, they consist of the building's shape (or dimension), default configuration parameters of the model, default state of a cell, etc. Each cell in a scenario identifies which physical element it represents (e.g., walls, chairs, vents, or furniture in a building) and its initial state. The base scenario is available in various formats like image, PDF, and BIM. We focus on BIM as it is available in 2D and 3D formats with the details of all the physical elements needed for simulation.

2.2 BIM

BIM is a method for three-dimensional object-oriented modeling that characterizes the geometry, spatial relationships, properties of elements in buildings, and cost estimates. It is mainly used for 3D visualization, clash detection, and construction review in the AEC (Architecture, Engineering, and Construction) industry. Implementation of BIM provides efficiency in real-time collaboration of projects in the AEC industry. BIM plays an important role in the life cycle of construction projects, as it shares the information of any physical object in the form of a digital format that is a factual basis for decisions. BIM can make construction drawings visible to non-professionals to understand the project (Wu et al. 2019). It produces data-rich models in construction projects that have helped the AEC industry improve its vision to the realization (Volk, Stengel, and Schultmann 2014). BIM-based data visualization can enhance the analysis of projects in many research domains like sustainable development, structural safety diagnosis, or building performance analysis. BIM models are suitable for visual communication to understand the project better. Hence, having access to and sharing of BIM with various end-users like modellers may help in a more realistic evaluation of building performance.

Often, the 3D models of the buildings are shared using an Industry Foundation Class (IFC) file that requires domain expertise for extracting data. IFC (IFC 2022) is an open standard file for exchanging BIM data. IFC files can generate the scenario for the DEVS simulator. It mainly follows two processes: Element Filter and Coordinate Extractor. Element filter is filtering and querying the required elements such as IFC Wall (for walls) or IFC Door (for doors) using an open API BiMServer (2022). It returns the coordinates of the filtered BIM in the form of x, y, and z coordinates. Coordinate Extractor builds a floor layout based on the filtered coordinate information that determines the size, the boundary of a building, and the number of cells in the model.

BIM provides realistic data and is practical to solve designs of complex buildings and simulation problems using the Cell-DEVS formalism. Likewise, visualizing the results of Cell-DEVS simulations provides contextual visualization. Using BIM in combination with Cell-DEVS to simulate various scenarios can help in predicting the building's performance at various stages. By analyzing and refining their ideas, these situations will help designers in making a conclusion.

In order to integrate BIM with Cell-DEVS, modelers need tools that allow them to access the BIM models, extract BIM metadata for Cell-DEVS simulations, and embed the results in BIM for contextual visualization. Recent cloud technology in the AEC industry has made BIM available for all industry stakeholders, including modelers. Many 3D modelling tools such as Autodesk 3ds Max, Blender and Maya are now available, through which users can access different BIM models.

2.3 Integration of DEVS into BIM

This section reviews some of the research done for integrating and visualizing the DEVS simulations. In their work, researchers have provided some methods to visualize the DEVS simulations, and a few of them use BIM. Still, no tool can integrate various DEVS models and simulators for the data visualization into BIM using a single platform.

(Muzy et al. 2008) developed software using an object-oriented approach for visualizing the fire spread DEVS model in the 3D. It uses bitmaps to create realistic effects (Muzy et al. 2005). (Hammad and Zhang 2011) create a 3D environment using Softimage (Autodesk Softimage 2022) to visualize the Cell-DEVS model. The model reviews the construction performance conflict analysis and visualizes the worksite in real-time. Ubisense software captures the real-time location data to transfer the data into Softimage. VITASCOPE (Kamat 2003) (Visualization of Simulated Construction Operations) is a user-extensible 3D animation language for visualizing construction operations. VITASCOPE uses a simple parametric-text animation description language to run discrete-event simulation tools. The sequential instructions create a 3D world that considers time, space appearance, and the interaction of people, machines, and materials during building constructions.

In (Freire, Wang, and Wainer 2013), the moving entities from the evacuation and occupancy Cell-DEVS simulation were visualized using Autodesk 3ds Max, a 3D modelling and visualization software. A custom built plug-in was developed for visualizing BIM-based Cell-DEVS models. This plug-in was written in MAXScript (an object-oriented built-in scripting language), and it was designed specifically for evacuation models. The script needs to be modified each time for integrating different DEVS models.

The research presented in (Castonguay, Pendergast, and Wainer 2009) focuses on integrating DEVS models developed with the CD++ simulator and Blender for 3D visualization. Blender supports python scripting to read and animate the simulation results. The visualization is integrated for an aircraft emergency evacuation model focusing on the delays of the passenger to exit the aircraft. The script supports the integration of visualization results for that particular model.

The DEVS Web Viewer (St-Aubin and Wainer 2022) is a web-based application to visualize and analyze DEVS, and the Cell-DEVS models. This tool supports the visualization of the simulation logs generated from various DEVS simulators. This web-based application, developed using JavaScript and HTML5, has a user-friendly GUI. It has multiple features such as a drag and drop option to load simulation files, a play-back widget to see the results animation forward and backward and can capture the video of the simulation results. This viewer can show DEVS simulation, GIS visualization of DEVS spatial models, and graph-based simulation results. It allows the users to upload simulation logs generated from the DEVS simulators and convert them into a standardized format. The results are animated on the browser through the canvas as bitmap images.

The BIM to DEVS API (Rajus et al. 2021) uses the Autodesk Forge platform to integrate DEVS simulations into BIM. This API extracts the properties of buildings (such as walls, furniture, windows, or vents) of the BIM, and their coordinates to create DEVS scenarios for Cadmium simulator. This API converts simulation logs generated in text format to CSV to animate the simulation results using Point Cloud. The BIM to DEVS

API is an early prototype that aims to provide modelers with varied expertise to access and visualize the data.

Tools like 3Ds Max and Blender require domain knowledge and expertise for integrating DEVS simulations for visualization. The reusability of the tool is lacking as the visualization plug-in developed is specific to a single Cell-DEVS model. While the DEVS Web Viewer supports multiple simulators and DEVS models, it does not support 3D model visualization. Most of these tools are used only for visualizing the simulation results but there is no single application for data extraction and visualization. BIM to DEVS API considers data extraction and visualization in the same application but the API currently needs to be customized for different DEVS models.

This research focuses on building a BIM to DEVS API by integrating the software architecture of DEVS Web Viewer (St-Aubin and Wainer 2022), which can handle outputs from various models and simulators for data visualization. The proposed architecture is demonstrated with case studies using Cell-DEVS models executed using CD++ and Cadmium simulators

3 PROPOSED SOFTWARE ARCHITECTURE

In this section, we show how to integrate the simulation results from various Cell-DEVS models and DEVS simulators into BIM. Figure 1 illustrates the proposed software architecture and shows the workflow of integration using the Autodesk Forge Tool.

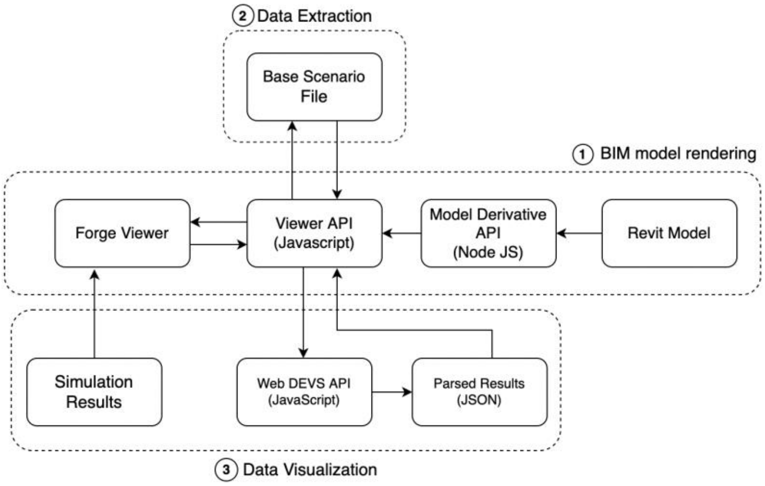


Figure 1**:** Software Architecture Diagram.

As shown, it is composed of three sub-components:

1. BIM model rendering - It shows the BIM model in a viewer using Autodesk Forge. The Viewer API fetches model data of a Revit model using the Model Derivative API and renders the BIM model in the forge viewer. This is discussed further in section 3.1.
2. Data Extraction - It generates the DEVS simulator's base scenario (initial value files) from the BIM model. Each BIM element's (x, y, z) coordinates are obtained using the Ray Intersection method. The file generated is used to integrate BIM into DEVS. Section 3.2 will include more details.
3. Data Visualization - It visualizes simulation results obtained from DEVS simulators into the BIM. It loads the simulation results, parses log files using the Web DEVS API, and displays them in the forge viewer using the Viewer API. This is explained in detail in section 3.3.

3.1 Rendering BIM model using Autodesk Forge

We use Autodesk Forge to access BIM models and integrate Cell-DEVS models into BIM, as stated in section 2.2. Users upload the models created using Autodesk Revit (Autodesk Revit 2022) to the viewer., These models are converted into files defined using Serial Vector Format (SVF – a general format for rendering model data in forge viewer), consisting of the BIM's boundary using the server-side Model derivative API built in NodeJS. The viewer API uses this information to render the 3D model in the forge viewer.

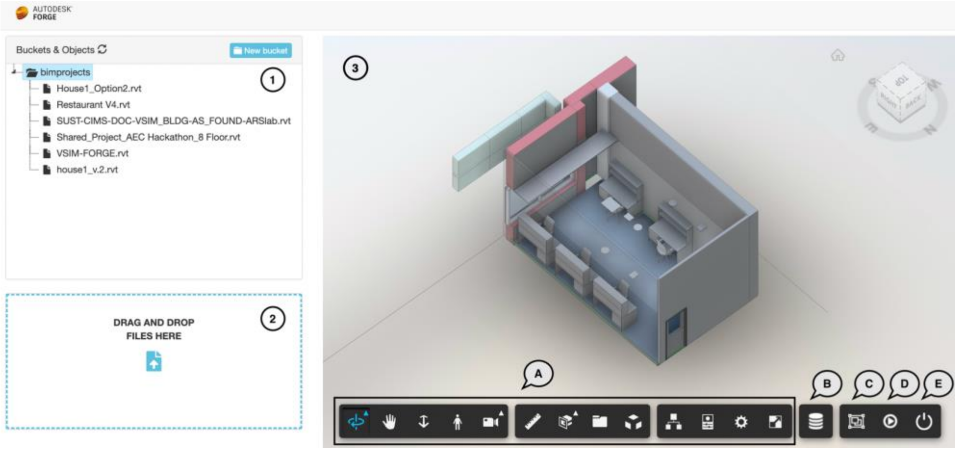
3.2 Data Extraction of BIM

The key idea of this module is to extract the geometries of the BIM and create a base scenario to be used for Cell-DEVS simulations. The BIM model can be accessed through the Viewer API after successfully rendering the model in the Forge Viewer. The data extraction through Forge can use multiple methods: bounding boxes, box colliders, ray-intersection, etc. In this research, we have used the ray intersection method to extract geometries of the BIM in Figure 1(2) using the Viewer API. This approach is mostly done on the client-side. This method is efficient and can extract complex BIM geometries (like slanting or curving walls) with ease. Various elements of the BIM can be easily differentiated by drawing the image using the ray intersecting points on the canvas. The first step is to generate a bounding box of a BIM using the bounding box method. Then, “rays” are passed from the origin and bottom of the bounding box containing the entire BIM model using the ray method (Ray 2022), after which, the ray intersection determines the objects by selecting their database IDs assigned in the Revit file to each element of BIM and a ray intersecting the area in the bounding box. The intersection of the elements generates the coordinates of the BIM geometries in a voxel format. These coordinates, their corresponding database IDs, and the type of BIM elements are parsed to convert into the grid, which creates the required base scenario for the Cell-DEVS model. Refer to (Rajus et al. 2021) for more details on different extraction methods.

3.3 Data Visualization of BIM

The proposed architecture includes a data visualization component (Figure 1 (3)) that allows users to integrate the results from various DEVS models (our implementation uses CD++ and Cadmium simulators). The simulation results obtained from these simulators can have different output formats; the simulation results are parsed into a more standardized format to increase readability and navigation. The Web DEVS API (St-Aubin et al. 2019) is a front-end client service used for parsing these varied results into an interoperable format that is smoothly read by the front-end application. This API considers the simulation log files from various DEVS models executed using different DEVS simulators. In our application, simulation results are passed to the Web DEVS API by uploading log files through the Viewer API. Three JSON files are generated from this API, one is the parsed simulation log file, another is the structure.json file (including information like which DEVS models or simulators are used in the execution), and an empty style.json file (to add colors for the animation).

These parsed data obtained is well structured and easily integrated into a Point Cloud, which adapts the results based on different models using the data from these three files from Web DEVS API through the Viewer API. First, the Point Cloud determines the coordinates of the cells from the simulation log file and draws the points in the BIM on the Forge viewer. The points are animated by updating the positions of the cells from the log file mentioned in the style.json file.



bimprojects
- House1_Option2.rvt
- Restaurant V4.rvt
- SUST-CIMS-DOC-VSIM_BLDG-AS_FOUND-ARSlab.rvt
- Shared_Project_AEC Hackathon_8 Floor.rvt
- VSIM-FORGE.rvt
- house1_v.2.rvt

Figure 2: Forge Viewer Interface for BIM to DEVS Integration.

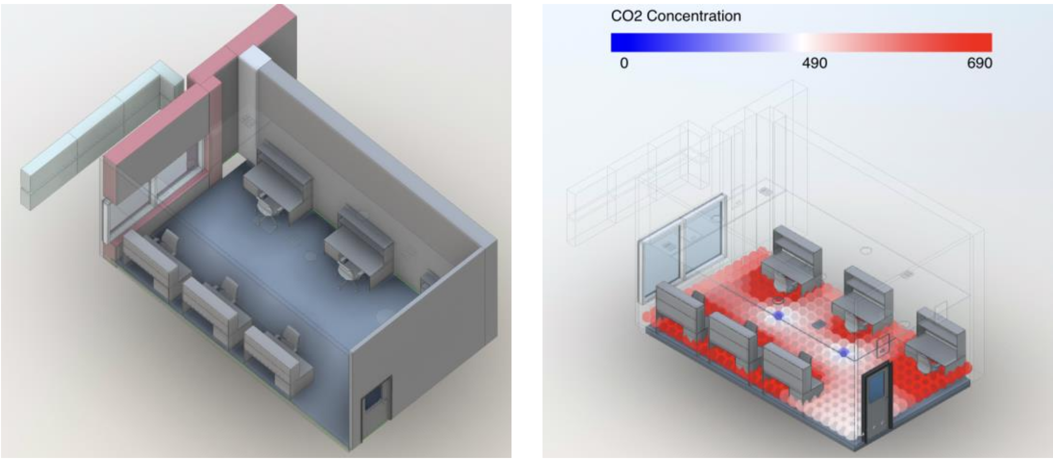
Figure 2 shows the Web Viewer Interface in Autodesk Forge; (1) shows a list of Revit BIM files (2) includes a drag and drop feature that allows users to upload or browse the simulation results generated from various DEVS simulators and (3) shows the Viewer Interface with the actual 3D BIM rendering, including various extensions of the Viewer settings (3B-E) besides the default services (3A). The custom extensions include the BIM Data Extraction (section 3.1 - 3B), 3C loads the simulation result files from the drop zone (Figure 2 (2)) in the forge viewer, then Viewer API send these files to the Web DEVS API, which reads and parses the simulation results. 3D is a play simulation extension that reads the parsed results and animates the simulation results through Point Cloud, and 3E is the color legend with the ON/OFF feature.

4 APPLYING THE ARCHITECTURE FOR INDOOR DIFFUSION MODELS

Based on the proposed software architecture, we introduce two case studies applying the workflow for simulation and visualization of two Cell-DEVS models into BIM. This section describes a CO2 diffusion model (Khalil and Wainer 2021) and indoor analysis for COVID-19 spread (Hajj-Ali and Wainer 2021) simulated with CD++ and cadmium simulators utilizing different BIM-based base scenarios. In the two examples presented in this section, we illustrate the process of integrating the indoor viral spread using the software architecture explained in Figure 1. The Carleton University lab workspace model was constructed with Autodesk Revit. This model is uploaded in the Autodesk Forge Tool and rendered using the Viewer API (1). The characteristics needed to simulate viral particles, such as the closed space dimensions, vent placement, tables and chairs, are extracted from the BIM models (2) to create a base scenario for the simulator. The Viewer API sends a request to the Web DEVS API, which parses the log files and sends the parsed results back to the Viewer API, which renders the results using point clouds in the BIM on the forge viewer (3).

4.1 CO2 Diffusion model using CD++ simulator

The system architecture was evaluated using a Cell-DEVS model of CO2 diffusion indoors in a Carleton University lab workspace (Figure 3a). The BIM for this lab has been extracted from Carleton University Digital Campus Innovation (http://cims.carleton.ca/#/projects/DigitalCampusInnovation). The purpose of this CO2 diffusion model is to determine the ideal sensor placement for accurate occupancy detection. Based on the presence of occupants and the ventilation system, the model represents an increase in CO2 concentration. This model helps in the study of building interior air quality by showing that a well-ventilated layout lowers CO2 distribution. This Cell-DEVS model is defined and executed using the CD++ simulator. The lab workspace has six workstations, each of which is occupied by research assistants (used as CO2 sources for the Cell-DEVS model), one window (that cannot be opened), one door, and two vents. The base scenario for the CO2 diffusion model simulation is based on this data.



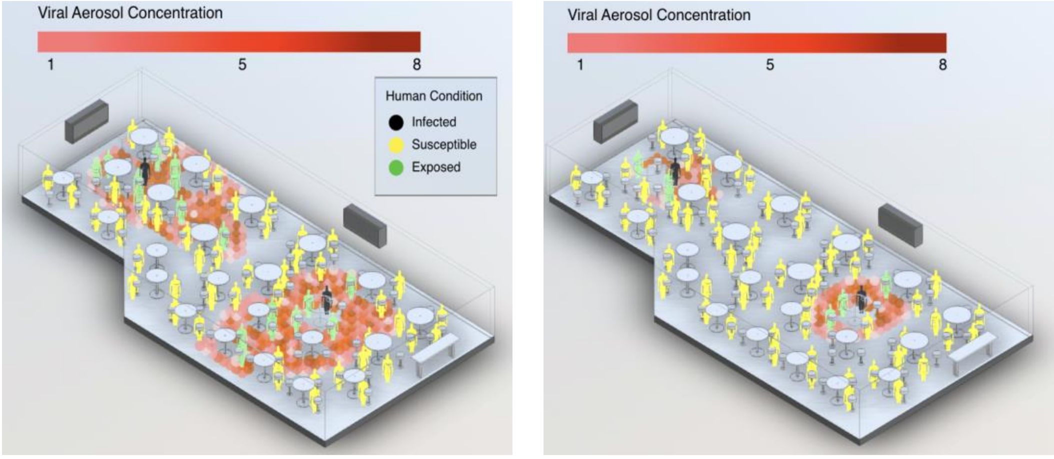
(a) (b)

Figure 3: (a) Lab workspace model and (b) CO2 diffusion spread in Lab.

The simulation results of the CO2 spread are shown in Figure 3b using the point clouds. The red color of the point clouds indicates an increase in CO2 concentration near the workstation since people there exhale CO2 every 5 seconds as they breathe. The color legend reflects the concentration level, with blue showing lower concentration and red indicating higher concentration.

4.2 Indoor Analysis for COVID-19 spread model using Cadmium

Figure 1 shows a workflow that was assessed using a Cell-DEVS model of indoor viral spread of COVID-19 in a restaurant. This is a well-known case of indoor spread of the disease based on poor ventilation and the influence of air conditioning in the restaurant. The model was built based on information from the actual study (Lu et al. 2020), which includes a floor plan of the restaurant that was used to create the BIM for the study. The Cell-DEVS model (Hajj-Ali and Wainer 2021) was first used to validate the model spread based on the actual results of the study and was later modified to illustrate the effectiveness of various mask types in reducing the spread of airborne viruses in an indoor environment. A small number of infected persons, as well as a number of healthy people, are placed in a room that produces the virus particle when they do some activity such as breathing, coughing, or speaking. The direction and speed with which the virus spreads indoors are affected by the ventilation system. This Cell-DEVS model is defined and executed using the Cadmium simulator.



(a) (b)

Figure 4: Indoor Viral Spread in a restaurant model (a) occupants wearing NO-MASK (b) occupants wearing COTTON MASK.

In this situation, the chairs are sources (individuals). To execute a simulation, a model file that defines the behavior of the viral spread and the base scenario extracted from the BIM is needed. It creates a log file that captures cell location, viral concentration levels, and individual infectious status over time. Figure 4 shows the viral concentration represented using point clouds and the infectious state of the individuals. Figure 4a shows the viral spread in which individuals are not wearing masks, whereas 4b shows the viral spread in which people wear cotton masks. The color legend shows the viral particle concentration, with light red signifying low concentration and dark red indicating high concentration. The person with the black color is infected (emits viral particles), the person with the yellow color is vulnerable (at risk), and the person with the green color is exposed (representing anyone present in a cell with viral concentration).

5 CONCLUSION

We introduced an interactive software architecture for integrating, visualizing, and analyzing simulation results in BIM using a single platform. The suggested method enables the integration of Cell-DEVS models in a variety of BIM situations. We demonstrated how the pre-design phase of a construction project may be eased by determining the indoor ventilation rate depending on the occupancy of people and the viral spread in a building by analyzing the simulation results under various BIM scenarios. This BIM architecture is not limited to the study of ventilation; it can also be used to research other DEVS models into BIM. We illustrated how to automatically extract information from a BIM, simulate the Cell-DEVS model, and view the results in BIM.

The proposed method is highly automated, allowing users to interact directly with the GUIs within BIM tools. Users simply need to utilize the built-in extensions to view their data. The proposed architecture supports a configurable mechanism for integrating various Cell-DEVS models executed using Cadmium and CD++. The DEVS model will be easily accessible and applicable for design analysis and to a variety of professional users as a result of this integration. Engineers, designers, and architects may use it to analyze and explore their building plans by using real-time simulations and making changes to the designs. These models may be easily applied to a variety of architectural contexts to see how contextual design influences simulation outcomes. The cloud platform and DEVS make it easier to collaborate on building simulations and access different building context models. In the future, we will expand the tools to allow remote model execution using cloud computing technologies. Improvements to the user interactions in the tool can be added in future for obtaining a better visualization and accessibility.

REFERENCES

Autodesk Inc. 2022. “Autodesk Forge Documentation”. https://forge.autodesk.com/developer/. Accessed Oct. 26, 2021.

Autodesk Inc. 2022. “Autodesk Forge Viewer”. https://forge.autodesk.com/en/docs/viewer/v7/. Accessed Oct. 26, 2021.

Autodesk Inc. 2022. “Autodesk Forge Model Derivative API”. https://forge.autodesk.com/en/docs/model-derivative/v2/reference/http/job-POST/. Accessed Oct. 26, 2021.

Autodesk Softimage 2022. https://autodesk-softimage.apponic.com/. Accessed Feb. 15, 2022.

Azhar, S. 2011. “Building Information Modeling (BIM): Trends, Benefits, Risks, and Challenges for the AEC Industry”. *Leadership and Management in Engineering* vol.11, pp. 241–252.

Belloli, L., D. Vicino, C. Ruiz-Martin, and G. Wainer. 2019. “Building Devs Models with the Cadmium Tool”. In *Proceedings of the 2019 Winter Simulation Conference*, pp. 45–59. National Harbor, Maryland.

Bergero, F., and E. Kofman. 2011. “PowerDEVS: A Tool for Hybrid System Modeling and Real-Time Simulation”. *SIMULATION* vol. 87, pp. 113–132.

BiMserver 2022. https://bimserver.org/. Accessed February 5, 2022.

Boccara, N. 2010. *Modeling Complex Systems*. Springer Science and Business Media, 2010.

Castonguay, P., T. Pendergast, and G. Wainer. 2009. "Tools for DEVS Modeling, Simulation and 3D Visualization". In *Proceedings of the 2nd International Conference on Simulation Tools and Techniques,* pp. 1-2. Rome, Italy.

Chidisiuc, C., and G. Wainer. 2007. "CD++Builder: An Eclipse-Based IDE for DEVS Modeling". In *Proceedings of the 2007 Spring Simulation Multiconference*, pp. 235–240. Norfolk,Virginia.

Concepcion, A.I., and B.P. Zeigler. 1988. "DEVS Formalism: A Framework for Hierarchical Model Development". *IEEE Transactions on Software Engineering* vol.14, pp .228–241.

Freire, V., S. Wang, and G. Wainer. 2013. "Visualization in 3ds Max for Cell-DEVS Models Based on Building Information Modeling". In *Proceedings of the Symposium on Simulation for Architecture and Urban Design*, pp. 1-8. San Diego, CA.

Hajj-Ali, Z., and G. Wainer. 2021. "Spatial Models and Masks in Indoor Analysis for the Spread of COVID-19". In *Proceedings of the 2021 Winter Simulation Conference,* pp. 1-12. Phoenix, Arizona.

Hammad, A., and C. Zhang. 2011. "Towards Real-Time Simulation of Construction Activities Considering Spatio-Temporal Resolution Requirements for Improving Safety and Productivity". In *Proceedings of the 2011 Winter Simulation Conference*, pp. 3538–3549. Phoenix, Arizona.

IFC 2022. https://www.buildingsmart.org/standards/bsi-standards/industry-foundation-classes/. Accessed March 27, 2022.

Jiang, Y., J. Ming, D. Wu, J. Yen, P. Mitra, J.I. Messner, and R. Leicht. 2012. "BIM Server Requirements to Support the Energy Efficient Building Lifecycle". In *Proceedings of International Conference on Computing in Civil Engineering*. Clearwater Beach, FL.

Kamat, V. 2003. *VITASCOPE: Extensible and Scalable 3D Visualization of Simulated Construction Operations*. Ph.D. thesis, Department of Civil Engineering, Virginia polyethnic Institute State University, Blacksburg, Virginia. Available via http://live.engin.umich.edu/wp-content/uploads/sites/430/2019/02/Kamat.PhDDissertation.pdf. Accessed Mar 27, 2022.

Khalil, H., and G. Wainer. 2021. "Modeling Carbon Dioxide Dispersion Indoors: A Cell-DEVS Experiment". In *Proceedings of International Conference on Cellular Automata for Research and Industry* vol.12599, pp. 226–236.

Kim, S., H. Sarjoughian, and V. Elamvazhuthi. 2009. "DEVS-Suite: A Simulator Supporting Visual Experimentation Design and Behavior Monitoring". *In Proceedings of the 2009 Spring Simulation Multiconference*. San Diego, CA.

Lu, J., J. Gu, K. Li, C. Xu, W. Su, Z. Lai, D. Zhou, C. Yu, B. Xu, and Z. Yang. 2020. "COVID-19 Outbreak Associated with Air Conditioning in Restaurant, Guangzhou, China, 2020". *Emerging Infectious Diseases* vol. 26, pp. 1628-1631.

Muzy, A., D. Hill, M. Joubert, and E. Innocenti. 2008. "A Post-Processed 3D Visualization Tool for Forest Fire Simulations". In *1st International ICST Conference on Simulation Tools and Techniques for Communications, Networks and Systems* vol. 41. Marseille, France.

Muzy, A., E. Innocenti, A. Aïello, J.F. Santucci, P.A. Santoni, and D. Hill. 2005. "Modelling and Simulation of Ecological Propagation Processes: Application to Fire Spread". *Environmental Modelling & Software* vol. 20, pp. 827–842.

Rajus, V. S., M. Patel, R. Cardenas, O. Kawach, and G.Wainer. 2021. "A cloud-based workflow for the integration of BIM to DEVS". In *Proceedings of the 2021 Symposium on Simulation for Architecture and Urban Design (SimAUD2021).*

Ray 2022. https://threejs.org/docs/#api/en/math/Ray. Accessed February 15, 2022.

Sarjoughian, H., and V. Elamvazhuthi. 2009. "CoSMoS: A Visual Environment for Component-Based Modeling, Experimental Design, and Simulation". In *Proceedings of the 2nd International Conference on Simulation tools and techiques*. Rome, Italy.

St-Aubin, B., E. Yammine, M. Nayef, and G. Wainer. 2019. “Analytics and Visualization of Spatial Models as a Service”. In *Proceedings of the 2019 Summer Simulation Conference*, pp. 1–12. San Diego, CA.

St-Aubin, B., and G. Wainer 2022. “ARSLab DEVS web viewer”. https://staubibr.github.io/app-simple/index.html. Accessed March 27, 2022.

Volk, R., J. Stengel, and F. Schultmann. 2014. “Building Information Modeling (BIM) for Existing Buildings — Literature Review and Future Needs”. *Automation in Construction* vol. 38, pp. 109–127.

Wainer, G. 2009. *Discrete-Event Modeling and Simulation: A Practitioner’s Approach (Computational Analysis, Synthesis, and Design of Dynamic Systems)*. CRC Press, Inc.

Wainer, G. 2002. “CD++: A Toolkit to Develop DEVS Models*”. Software: Practice and Experience* vol. 32, pp. 1261–1306.

Wainer, G., and J. Fernández. 2016. “Modelling and Simulation of Complex Cellular Models Using Cell-DEVS”. *Simulation* vol. 92, pp. 101–115.

Wu, Z., C. Chen, Y. Cai, C. Lu, H. Wang, and T. Yu. 2019. “BIM-Based Visualization Research in the Construction Industry: A Network Analysis”. *Interantional journal* of *environmental research and public health* vol.16.

Zeigler, B., H. Prähofer, and T.G. Kim. 2000. *Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems*. Academic Press. San Diego, CA.

AUTHOR BIOGRAPHIES

MITALI PATEL is a M.A.Sc student at Carleton University. She holds a bachelor’s in Electronics and Communication Engineering from GTU India. Her research interests lie in integrating the DEVS and BIM for visualizing the simulations in BIM. Her email address is mitalipatel3@cmail.carleton.ca.

VINU SUBASHINI RAJUS is a Post-Doctoral Fellow at the Department of Systems and Computer Engineering at Carleton University. She holds a Ph.D. in Interactive arts & technology from Simon Fraser University. Her research interests include sustainable design practices and human behavior, housing, energy modelling, interaction design, and data visualization. Her email address is vinusubashinirajus@cunet.carleton.ca.

GABRIEL WAINER is a Full Professor in the Department of Systems and Computer Engineering at Carleton University. He received the M.Sc. (1993) at the University of Buenos Aires, Argentina, and the Ph.D. (1998, with highest honors) at UBA/Université d’Aix-Marseille III, France. He is a Fellow of SCS, the Society for Modeling and Simulation International. His research interests include discrete-event simulation, parallel simulation, and Building Information Models. His email address is gwainer@sce.carleton.ca.