DEVS-BASED BUILDING INFORMATION MODELING AND SIMULATION FOR EMERGENCY EVACUATION

Sixuan Wang   Vinu Subashini Rajus
Michael Van Schyndel   Robert Woodbury
Gabriel Wainer

Dept. of Systems and Computer Engineering   School of Interactive Arts and Technology
Carleton University   Simon Fraser University
1125 Colonel By Dr. Ottawa   250-13450 102nd Avenue, Surrey
ON K1S5B6, CANADA   BC V3T0A3, CANADA

ABSTRACT

Nowadays, numerous Computer-aided Design (CAD) software packages support Building Information Modeling (BIM). BIM software can benefit of advanced simulation in the pre-design phase of construction projects. In this case, we show focus on models of the emergency evacuation regarding the security and safety. We analyze the evacuation simulation of a model based on DEVS (Discrete Event systems Specification) for BIM authoring tools. The idea is to automate to extraction of building information that can be subsequently used in a simulation. Our case study uses a Cell-DEVS model of the evacuation of a multi-floor building. We also show how to obtain a 3D visualization by transforming the simulation results, facilitating the work of architects, contractors and fabricators. This kind of application could be used to analyze bottlenecks and the maximum occupation for determining an optical evacuation plan.

1 INTRODUCTION

In recent years, architects and contractors have faced numerous new challenges when developing construction projects. Until recently, the success of the pre-design phase of a building construction was dependent on the experience of experts. However, nowadays, the information required for completing the project and the amount of data to be analyzed are larger and more complex. Consequently, new tools have been developed to support the pre-design phase directly, including the enhancement of Computer Aided Design software (CAD) with Building Information Modeling - BIM (Brad 2009). BIM tools provide a method of generating and managing building data using advanced support tools (Ham et al. 2008, Mihindu and Arayici 2008). BIM uses 3D real-time dynamic building modeling to achieve accurate construction projects with minimized costs, improving the way architects-contractors and fabricators work (Lee et al. 2008, Hardin 2009). BIM software creates parametric 3D models (instead of 2D perspectives) and operates on a digital database; hence, when any change is made, it is reflected in the whole drawing.

Current BIM applications have incorporated modeling and simulation (M&S) as a mechanism to improve the quality of the building design. Related tools include models for indoor climate, energy consumption and CO_2 emissions. However, they do not take into consideration other design issues. For instance, problems in emergency planning and evacuation have increasingly drawn the attention of experts in building design and related applications, as they deal with human safety and property security (Boukerche et al. 2009). M&S has been successful in studying such detailed phenomena by providing a constrained experimentation environment for better emergency management (Kincaid et al. 2003). Such evacuation models and the simulations aim to not only find evacuation plans of emergency exits in buildings, but also ensure a safe and efficient evacuation time of all people inside the buildings. Therefore,

studying the emergency planning in case of an emergency crisis is an important design step for successful construction projects (Boukerche et al. 2009).

A variety of methods has been employed for modeling and simulating this kind of applications. In this research, we use the Discrete Event System Specification (DEVS) formalism (Zeigler et al. 2000), which defines a theory for discrete-events systems specification. DEVS M&S provides a framework for building models composed by behavioral (atomic) and structural (coupled) components. Cell-DEVS (Wainer 2009) extended DEVS by allowing building cellular models (in which each cell is a DEVS atomic model, and the cell space is a DEVS coupled model). Cell-DEVS has shown to avoid unnecessary processing in inactive regions by using a continuous time base, and it provides a simple mechanism to model timing delays. We have explored a new method to integrate advanced BIM software with formal models built in DEVS and Cell-DEVS. We introduce the method, and show its application to study evacuation processes, which can be used to understand the bottlenecks and congestions during the building design phase. The interactive simulation system that we present here provides an advanced emergency model in discrete-event cell spaces and a seamless integration of BIM and Cell-DEVS environment, taking full advantage of each tool's strength. We use CD++ (Wainer 2009) for Cell-DEVS, and Autodesk Revit Architecture (AutoDesk 2012b) and Autodesk 3ds Max for BIM (AutoDesk 2012a). We will show how to automatically extract the building information from BIM (including the information from multiple floors), using it to modify a Cell-DEVS model, which can be used to study the evacuation of people under different design scenarios. Likewise, we show how to perform a 3D visualization by transforming the simulation results back into BIM, which can be used for validation and analysis of different scenarios. Furthermore, the method for interactive simulation provides an easy and fast reconfigurable mechanism for the simulation properties (e.g. the number and location of doors, the maximum occupancy, the floor layout, etc.). By investigating the visualization simulation results under different scenarios, it can facilitate the pre-design phase for finding evacuation plans within satisfactory consuming time, avoiding congestions and risks. Moreover, if any problems related to the evacuation process arise, they can be easily fixed instead of trying to make changes after construction has begun.

Section 2 of this paper discusses the related work in building information modeling and simulation and main differences of our work. Section 3 represents the interactive system architecture, followed by detail descriptions of each sub-system with a building example. Section 4 illustrates and analyzes the visualization results of the evacuation model. Finally, we conclude this work in Section 5.

2   RELATED WORK

BIM uses 3D, real-time, dynamic building modeling software to increase productivity in the building design and construction (Ahmed et al. 2010). At present, there are numerous BIM software applications, including Autodesk Revit Architecture (AutoDesk 2012b), Autodesk 3Ds Max (AutoDesk 2012a), Bentley Architecture (Bentley Systems 2012), Graphisoft ArchiCAD (Graphisoft 2012), VectorWorks Architect (VectorWorks 2012) and others. They all provide powerful mechanisms for BIM and 3D visualization, facilitating the workload of building design, improving construction projects and minimizing financial costs (Mihindu and Arayici 2008). BIM software is often built on the Industry Foundation Classes -IFCs-(BuildingSMART 2012), an open standard for exchanging BIM data in the building industry.

Modeling and simulation has been applied in construction projects for analyzing the building process and helping decision-making. Some of the most popular software in this domain are MicroCYCLONE (Halpin 1977), Simphony (Hajjar and AbouRizk 1999), and Stroboscope (Martinez 1998), which have been proved to be effective and efficient in simulating logical relationships between different resource locations. Cell-based analysis has been used in the construction site analysis and emergency simulation. Elbeltagi et al. (2004) represented a space representation of work zones for resources management with small grid units that take irregular shapes. Other works like (Pelechano and Malkawi 2008, Yang et al. 2005) provided different simulation models for building evacuation based on cellular automata.

In recent years, we have developed different construction and architecture projects with DEVS and Cell-DEVS. In (Zhang et al. 2007, Hammad et al. 2011) the authors introduced Cell-DEVS models for

construction sites, trying to deal with the construction performance and the crane behavior that enable the conflict analysis and visualization of work site. In (Ahmed, Wainer, and Mahmoud 2010), an Interactive Environment System (IES) was introduced for a basic integration between BIM and Cell-DEVS, in which a simulation of Diffusion Limited Aggregation (DLA) was used to model the growth of mold in building walls. In (Wainer et al. 2007), a Maya/CD++ tool was presented for integrating the Autodesk Maya 3D visualization (AutoDesk 2012c) and Cell-DEVS evacuation models built in CD++, based on the emergency evacuation process. However, most of these efforts required ad-hoc tailoring and they provide simple test scenarios that are not adequate for studying emergency evacuation and other advanced simulations in BIM authoring tools.

Ruppel and Abolghasemzadeh (2009) raised three main challenges of the emergency planning: 1) virtual simulation environment based on boundary conditions, 2) mutual interaction between parameters, and 3) realistic visualization of the simulated results. In (Eastman et al. 2009), the authors developed a rule checking system for evaluating the evacuation performance, and proposed a concept of a synthesis process to use sharable BIM representations and evacuation simulations based on rule checking tools. In (Ruppel and Abolghasemzadeh 2009), a BIM-ISEE (Immersive Safety Engineering Environment) was developed, focused on the realistic virtualization in their immersive environment for integrating fire and evacuation simulation with BIM tools. Nevertheless, these tools lack of the generality of DEVS/Cell-DEVS M&S, which can also be applied to other problems.

In the following sections, we introduce an advanced interactive simulation system for integrating BIM software and DEVS tools, facilitating the work of pre-design phase of construction projects when concerning emergency evacuation. The current research:

- Combines BIM authoring tools, Cell-DEVS and visualization. A component-oriented approach provides reusability and interoperability for the evacuation process within constructional building design.
- The interactive data collecting process with BIM provides higher automatic and comprehensive mechanism to get directly input information for CD++.
- The emergency evacuation model supports multiple level floors based on Cell-DEVS, which provides wider feasibility and better complexity when taking account to stairs occupancy.
- The 3D visualization uses preprocessing to parse simulation, accelerating the visualization process by using the visualization interface that can be customized easily with MAXScript.

3 INTERACTIVE SIMULATION SYSTEM ARCHITECTURE

Figure 1 illustrates the interactive simulation system architecture. As we can see, it is composed of three sub-systems: BIM Data Collection, DEVS Simulation and BIM 3D Visualization.
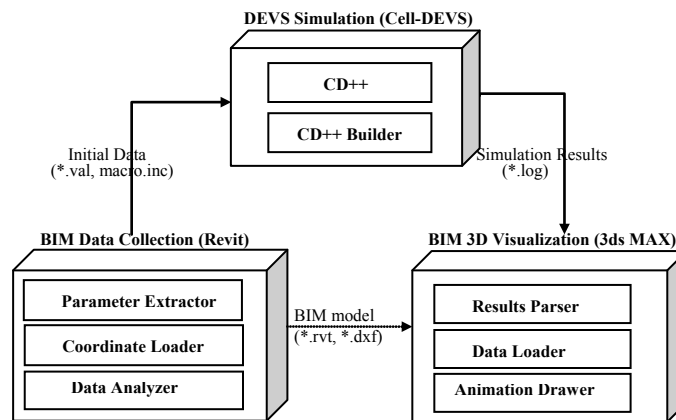

Figure 1: Interactive simulation system architecture

This component-oriented architecture can be easily reused and extended to other BIM tools, as long as the selected BIM tool supports the interface of the system architecture.

- **BIM Data Collection**: it is done automatically, generating initial data files for the DEVS simulator. This includes a parameter extractor for various properties, and a loader for getting (x, y, z) coordinates of each BIM element. The generated files are used for integration of BIM and DEVS models.
- **DEVS Simulation**: it builds a simulation model based on the BIM data collected, and it then executes simulation in CD++. The simulation results can be seen in 2D scenes using CD++.
- **BIM 3D Visualization**: it visualizes the simulation results in 3D, providing a more intuitive means for analysis. It parses the simulation log files, and it loads the results and BIM file into the BIM software. The simulation is generated by the animation drawer.

We used Autodesk Revit Architecture (AutoDesk 2012b) for the BIM Data Collection, and Autodesk 3Ds Max (AutoDesk 2012a) for BIM 3D Visualization (as it includes powerful 3D effects, animation and rendering). As a case study, we will present different simulation scenarios in an office building with two floors and an open terrace (presented in Figure 4), including stair cores on either end of the building for circulation. Floors 1 and 2 have cubicles, enclosed office spaces, and other basic amenities. Elements like the stair entries and exit doors, furniture were considered for modeling evacuation, and different scenarios were analyzed in order to understand bottlenecks and congestion under emergency evacuation.

3.1 Data Collection

BIM software provides a large amount of information for a building; in order to get the set of data to initialize a CD++ simulation, we need to define a data collection strategy. Autodesk Revit Architecture was used for this purpose, as it includes a well-defined parametric API and a BIM add-in. Revit can be used to build 3D models and 2D drawings, and the Revit API can be used to find and manipulate different model elements (floors, walls, doors, furniture, etc.). The Data Collection subsystem sends data to be simulated and visualized in CD++. The left part of Figure 2 illustrates the process structure of the subsystem. As we can see, this involves two phases: 1) collecting necessary data from BIM by *Extract Parameter* and *Load Coordinates*; 2) generating initial data files for Cell-DEVS by *Data Analyze* and *Set Macro*.
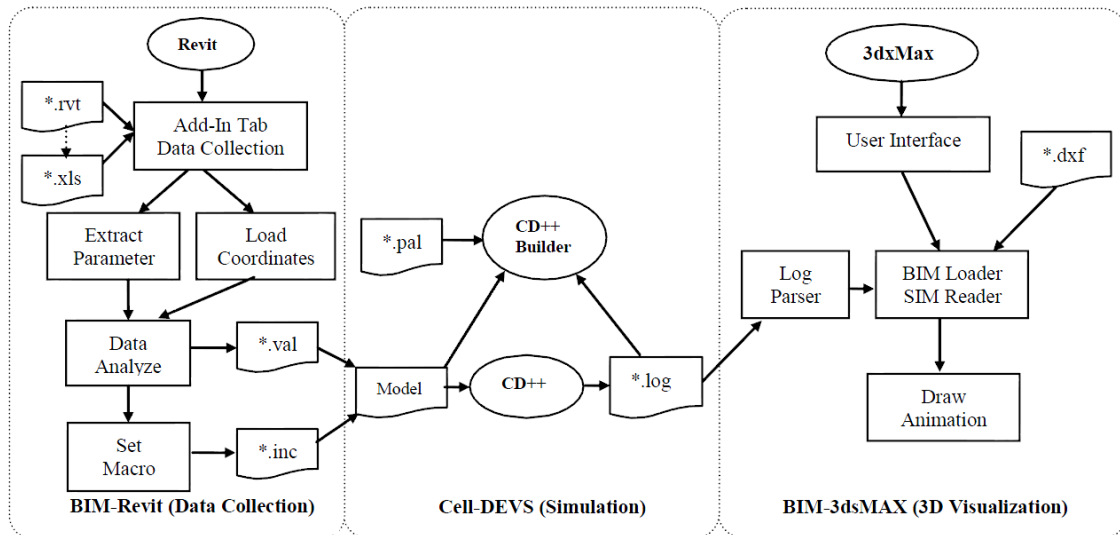

Figure 2: Data collection, simulation and visualization process.

In the first phase, we parse the BIM information stored in Revit files (*.rvt), and we extract the planar layout data of each floor, including the element type (e.g., wall, door, stair, furniture) and their corresponding coordinates. We can extract different model parameter data (e.g., type, length, width, color of

each element) and related model graphical data (e.g., coordinates, direction). Since an Element ID can identify each element, we select the entire element IDs of needed element types for each floor, and then we extract the parameters of the selected elements via the Revit API. We then find the corresponding coordinates with the same element IDs. Figure 3 shows a segment after the first phase, in which each line represents an element with the same type (*wall*) in a given floor (*Floor 1*), and the extracted parameters (Element ID, coordinates, width, length and area scale).

```
--Extract parameters & load coordinates--
168940: x(-724") y(244") o(0) width(8") length(672") area(676.00 SF)
168960: x(-724") y(244") o(1) width(8") length(144") area(136.00 SF)
168979: x(-580") y(340") o(0) width(8") length(576") area(492.00 SF)
169086: x(-580") y(340") o(1) width(8") length(72") area(72.00 SF)
169115: x(-508") y(576") o(0) width(8") length(240") area(724.00 SF)
169360: x(-236") y(340") o(1) width(8") length(812") area(812.00 SF)
169415: x(-724") y(-436") o(2) width(8") length(1299") area(1272.48 SF)
170658: x(-716") y(-236") o(1) width(8") length(144") area(144.00 SF)
```

Figure 3: Result segment of collecting phase.

In the second phase, we generate specific initialization data files *(*.val, *.inc)* for the CD++ simulation. The **.inc* file includes macros containing the data extracted from Revit. For instance, in our evacuation models, we first calculate the scale of each floor, converting it to a unified size: XMIN is the horizontal ordinate of the left-most point, while XMAX represents the horizontal ordinate of the right-most point. Similarly, we get YMIN and YMAX for the vertical ordinates. Then, knowing the size of each cell, we can compute the number of cells in each of the three dimensions, and generate the CD++ Macros.

```
#BeginMacro(XMIN) -724 #End        #BeginMacro(XMAX) 1868 #End
#BeginMacro(YMIN) -436 #End        #BeginMacro(YMAX) 578 #End
#BeginMacro(Cell_SIZE) 24 #End     #BeginMacro(Cell_X) 109 #End
#BeginMacro(Cell_Y) 43 #End        #BeginMacro(Cell_Z) 3 #End ...
```

The **.val* file contains initial values of each cell of the model in the simulation, according to the parameters and coordinates collected from the first phase. The following excerpt shows the result after the second phase. Here, (22, 26, 0) = 0 means that the cell is empty. Likewise, other element types in the model include walls (1), furniture (1), stair to down (13), stair to up (11), and exits (2), in accordance with the cell states listed in Table 1. The basic idea of initializing state of each cell is to "pave" all needed elements according to their type priorities. For each type following the priority order, we find all elements belonging to this type (in the current case, the order is wall, furniture, stair, door). For each element, we calculate all passing coordinates according to its starting coordinates and properties (e.g., length, width); finally, we assign the state value to the cells that hold these coordinates.

Based on the data collected, we build a model file for CD++ with the given rules (section 3.2). When the simulation is executed, a log file is generated. This simulation output can be used for 2D visualization.

This process can be automated directly inside the BIM tools. The current implementation used the Revit 2012 API for this subsystem, as seen in the right part of Figure 4. The functionality was encapsulated as components with configurable default variables. Although Revit provides a mechanism for carrying out these tasks, in order to realize the fully automated collection of needed coordinates (and to allow our work to be extended to other software tools), we need to use IFC (BuildingSMART 2012), an open standard that supports various tools for sharing, analyzing and editing BIM.

3.2 Evacuation Model

The multi-level evacuation model defined in CD++ is defined using a set of rules indicating the output value for the cell's state after satisfying a precondition. We divided the building into individual cells whose behavior depends on its current state (shown in Table 1).
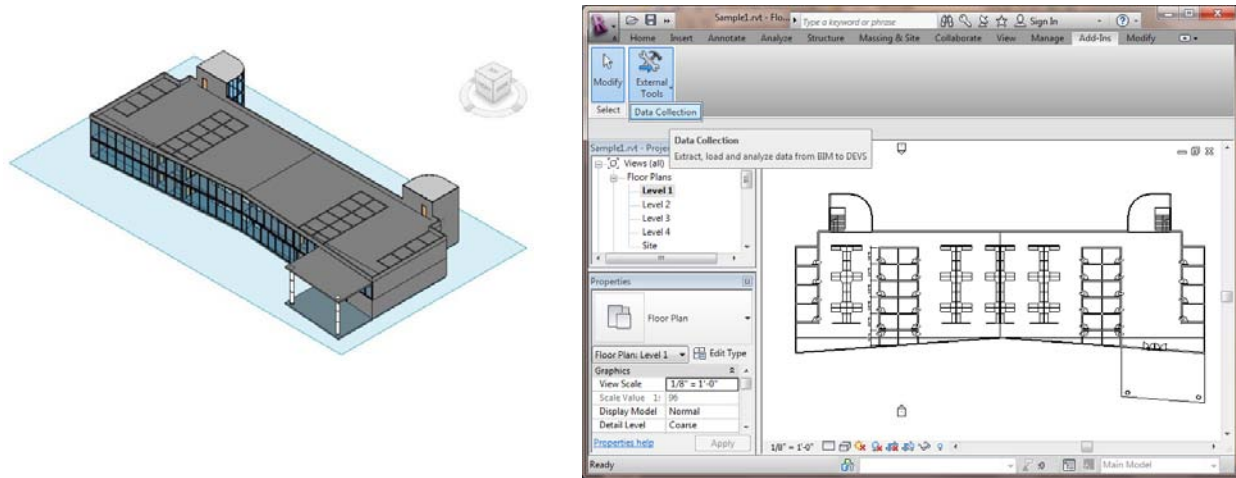
Figure 4: Building sample (left) and Revit user interface (right) in Revit.

Table 1: Cell states and definitions.

Cell State	Cell Color	State Name	Cell State	Cell Color	State Name
1		Wall	8		Up Occupied
2		Exit	9		Left
3		Down	10		Left Occupied
4		Down Occupied	11		Top of Stairs
5		Right	12		Top Occupied
6		Right Occupied	13		Bottom of Stairs
7		Up	14		Bottom Occupied

The model file contains information about the cell space dimensions, border behavior and neighborhood shape, this information is generated by the Data Collection subsystem. In our example, the model has space dimensions of 109×43×3 cells, and the borders have different behavior (representing people that left the building). The definition of the neighborhood deals with moving people in different floors. First, to prevent collisions, we use a neighborhood as in Figure 5.
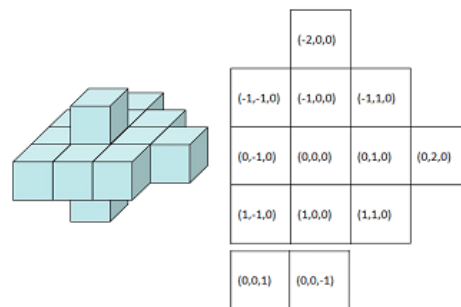

Figure 5: Neighborhood definition.

For the purpose of this evacuation model, the priority or the right of way is assigned to certain movement types in this order: down, left, up and then right. Two additional upper/lower stair cells, i.e. (0,0,-1) and (0,0,1), are needed to allow people to traverse from one floor to the next. The first step implementing the model is to overlay a Voronoi diagram of the route to an exit. Figure 6 demonstrates how a person follows the pathway to exit the building.

The next step involves modeling the actual movement of people. This can be broken into 5 subsections, each of them dealing with the different movements in a cell: 1) someone enters the current cell; 2) someone moves from the current cell to another one, 3) someone exits the floor, 4) someone enters a stairwell, and 5) someone exits a stairwell.
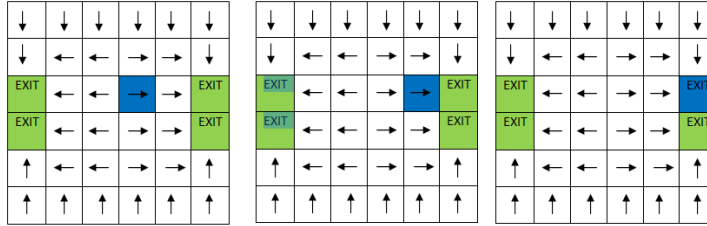

Figure 6: Pathway initialization.

The rules if someone entered an unoccupied cell are as follows:

```
rule : 4  100 { (0,0,0) = 3 and ( (0,1,0) = 10 or (-1,0,0) = 4 or (0,-1,0) = 6 or
        (-1,0,0) = 14 or (1,0,0) = 14 or (0,1,0) = 14 or (0,-1,0) = 14 )}
rule : 6  100 { (0,0,0) = 5 and ( (1,0,0) = 8 or (-1,0,0) = 4 or (0,-1,0) = 6 or
        (-1,0,0) = 14 or (1,0,0) = 14  or (0,1,0) = 14 or (0,-1,0) = 14 )}
rule : 8  100 { (0,0,0) = 7 and ( (1,0,0) = 8 or (0,1,0) = 10 or (0,-1,0) = 6 or
        (-1,0,0) = 14 or (1,0,0) = 14  or (0,1,0) = 14 or (0,-1,0) = 14 )}
rule : 10 100 { (0,0,0) = 9 and ( (1,0,0) = 8 or (0,1,0) = 10 or (-1,0,0) = 4 or
        (-1,0,0) = 14 or (1,0,0) = 14  or (0,1,0) = 14 or (0,-1,0) = 14 )}
```

This section is straightforward. If there is a person in the neighborhood (occupied cell) who needs to move into this cell, then, the person moves. The rules handles people leaving a cell are as follows:

```
rule : 3  100 { (0,0,0) = 4 and odd(1,0,0) }
rule : 9  100 { (0,0,0) = 10 and odd(0,-1,0) and (-1,-1,0) != 4 }
rule : 7  100 { (0,0,0) = 8 and odd(-1,0,0) and (-2,0,0) != 4 and (-1,1,0) != 10 }
rule : 5 100 {(0,0,0)=6 and odd(0,1,0)and (-1,1,0)!=4 and (0,2,0)!=10 and (1,1,0)!= 8}
```

These rules avoid collisions: if two cells have individuals that need to move into the same space, only one will move (and the other will wait for its turn). In this case, a movement to the right has the least priority. Therefore, when the current cell (0,0,0) is occupied by a 6 it must first ensure that no other cells are occupied by a value which would correspond to a movement to cell (0,1,0); that means (-1,1,0) cannot be moving down, (0,2,0) cannot be trying to move left and (1,1,0) cannot be moving up. The values of these movements would be 4, 10 and 8 respectively. For this reason, two extended neighborhoods were used.

The remaining rules are similar and follow the principles and layout as the previous rules. When entering, exiting or descending a stairwell or exiting the building through the exit door, they ensure that no collisions occur. Finally when exiting the building the person must "disappear" from the simulation:

```
rule : 9 100 { (0,0,0) = 10 and ((0,-1,0) = 2 or (0,-1,0) = 11) }
rule : 14 100 { (0,0,0) = 13 and (0,0,1) = 12 }
rule : 12 100 { (0,0,0)=11 and ((-1,0,0)=4 or (0,-1,0)=6 or (1,0,0)=8 or (0,1,0)=10 )}
rule : 13 100 {  (0,0,0)=14 and ((1,0,0)=3 or (1,0,0)=5 or (1,0,0)=7 or (1,0,0)=9 or
...
```

3.3   3D Visualization

The 3D visualization subsystem provides a method to process CD++ log files, and a GUI to interact with 3Ds Max (Figure 2). This sub-system contains two phases: 1) parsing the simulation results to get necessary data; 2) visualizing the parsed simulation file in user interface in the 3dx Max. We first read the log file through looking for simulation times, cells locations and values. A second phase provides a graphical user interface written in MAXScript to visualize the parsed log file in a 3D visual environment scene of

the BIM model. This BIM model is exported as a *.dxf file, and then imported into 3Ds Max. After loading the BIM model, we read the Cell-DEVS model (which contains the definition of behavior and initial data of Cell-DEVS models) and the parsed log file. Based on this, we draw the simulator data in the 3Ds Max scene. A GUI written in MAXScript enables the user to visualize the results, as shown in Figure 7.
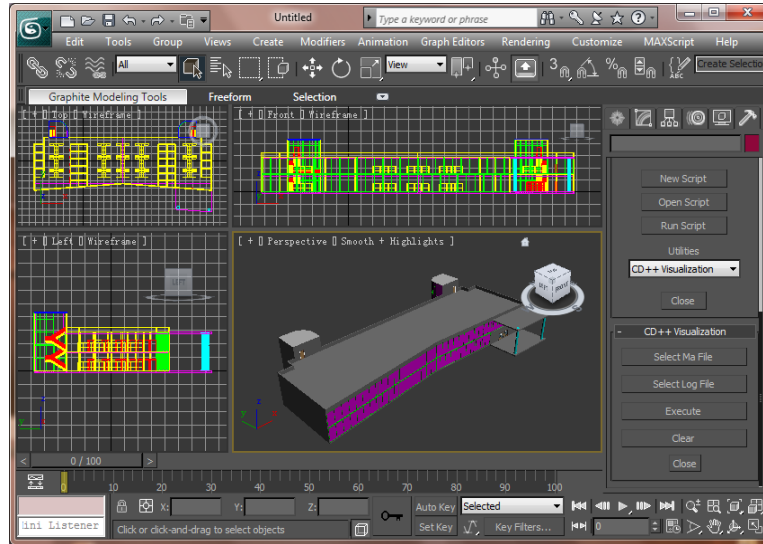

Figure 7: 3ds Max visualization user interface.

4   CASE STUDY: SIMULATION RESULTS

In this section, we discuss some simulation results, in order to find bottlenecks and maximum occupancy. They are generated from the evacuation model defined in Section 3, using different building designs (building structure, doors number, exit location, etc.). Our initial test (see Figure 8) shows a simple building model to find potential flaws in the building layout. This building contains four floors connected by a single stairwell (the upstairs are yellow cells and the downstairs are red ones). There are exit doors on the first floor (in green; dotted area). People try to go down along the arrow directions and finally get out the building through exits. At first, the building space is approximately 50% occupied (each blue cell represents one person). Each second, according to the rules, each person tends to move forward one cell following the shortest pathway towards downstairs/exits. It took 584 seconds to evacuate all the people.

Figure 8: Simulation results of a simple building at different simulation times.

To study the impact of door location/number in the evacuation time, we discuss two modifications to the original design (see Figure 9). The first one considers changing the exit locations: we moved the main exit of the first floor (which, originally, was in the middle of the leftmost wall). This modification resulted in shorter evacuation time (464s under occupancy of 50%). The potential reason of this difference is that it balances the evacuation process more evenly, considering new people coming from stairs on one side and exiting people on the other side. The result indicates that different locations of doors determine different evacuation times.

The second test in Figure 9 shows the impact of the number of doors. We modified the model again, and added two additional staircases connecting the second floor and first floors. The idea is to test if this results in a more rapid evacuation of the building. However, including more stairs makes the evacuation slower (642 s; worse than any other previous designs). The simulation visualization allowed us to identify the main cause of the problem: a bottleneck on the first floor in front of the main stairwell. In the previous design, for each floor, the people coming into it are approximately equal to the people leaving from it, which made the evacuation process execute smoothly at a constant rate. However, when additional staircases were added, the people on the second floor moved down to first floor faster, resulting in a massive bottleneck that prevented people from exiting at an optimal rate. As we can see, when people are entering the first floor from two different locations, they have to fight for the priority to exit the building.

Changing exit Locations

Adding more stairs

Figure 9: Simulation results of two modified building (exit location changed / more doors added).

The same model was used for determining the building's potential occupancy levels. Figure 10 shows different simulations runs for each of the building design mentioned above. The occupancies were set from 5% to 50% for demonstrating how to determine the potential maximum allowable occupancy given a safe evacuation time constraint. As seen from Figure 10, as the occupancy increases, the evacuation process requires more time with a roughly linear trend. It is reasonable because people evacuating at a considerable constant rate. Nevertheless, as seen in previous test results, due to differences of layout (e.g., number of door, exit location), the final evacuation time would differ. For example, the difference of the two modified tests (added stairs vs. changed location) with occupancy of 50% is nearly 200s, about a 40% increase. However, it is worth to note that this solution does not guarantee to be the optimal one.
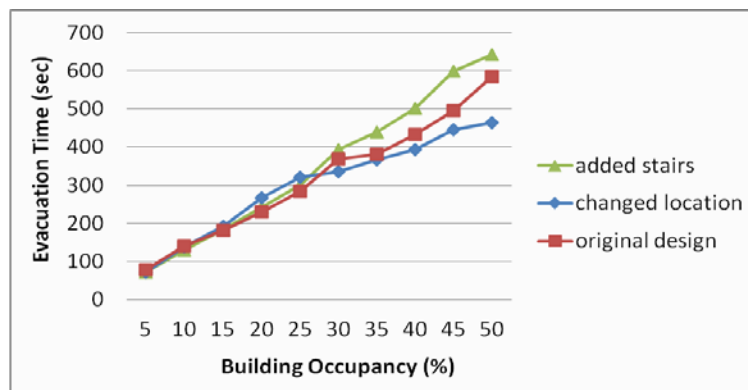

Figure 10: Building Occupancy vs. Evacuation Time.

So far we have shown that our evacuation model is capable of recreating an evacuation event and demonstrated its potential uses for testing building designs. We now show the evacuation of the actual BIM-based model mentioned in the previous sections using our developed 3D visualization GUI (see Figure 11). The main reason for using 3D rather than 2D is that it is more intuitive and attractive.

Assume that in real case, the normal occupancy level is about 20% (approximately 200 people); we want to know whether the current building design can fulfill the evacuation time constraint (e.g., 4 m) once an emergency happens. To do so, we use Revit to generate the data (Data Collection sub-system), and then the data is loaded into CD++ to execute the simulation. Once the log file has been parsed and loaded into 3Ds Max, the simulation results can be visualized using the GUI we developed, as showed in Section 3. Figure 11 shows what the visualization results for such case (it finally took 208 seconds to evacuate the building, which satisfies the time constraint 4 m).

a. (t = 00:00:001:000) *b. (t = 00:00:100:000)*

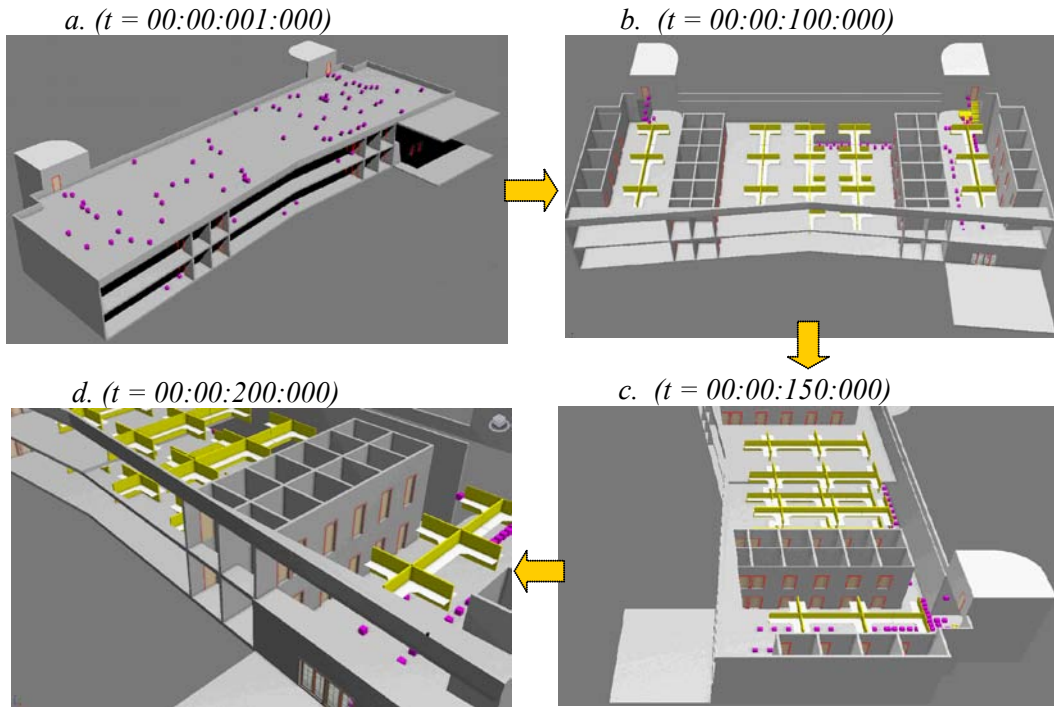*d. (t = 00:00:200:000)* *c. (t = 00:00:150:000)*

Figure 11: Different perspectives 3D simulation using 3Ds Max.

Figure 11 also illustrates the ability for the designer to see different perspectives within the simulation process, which is another benefit of using 3D. For example, a) shows a side view, b) an inside view, c) spinning, and d) zoom in/zoom out. By checking the simulation results, we can say that the building design satisfies the evacuation management requirements. With the evacuation plan in this model (see Section 3.2), related measures could be taken and put in place. Additionally, if the evacuation time is deemed too long or there would be more people in reality, the designers of the building would have to make adjustments to decrease the time it takes.

5   CONCLUSION

We presented an interactive simulation system by integrating BIM and DEVS models, with a study case for the advanced evacuation planning in a multi-level building. The basic idea is to take advantage of each tool's strength and let them collaborate efficiently. The prototype solution uses the CD++ toolkit for Cell-DEVS, and Autodesk Revit Architecture and Autodesk 3ds Max toolkits for BIM. We have shown the method to extract data automatically from BIM, to simulate the evacuation model and to visualize the results back in BIM. The whole process support considerable automation, directly interacting with our provided GUIs inside BIM tools, users only need to simply click few buttons or easily customize their own special requirements.

Furthermore, the presented system also provides an easy configurable mechanism for fast modifications of simulation properties. By investigating the 3D simulation results under different scenarios, we

showed how it could ease the pre-design phase of construction project to understand the bottle-necks/congestion and determine the maximum occupation within a satisfactory evacuation time. On the other hand, this work is not limited to the emergency planning; the results further confirm the potential feasibility of extending other DEVS models in this interactive BIM simulation application.

Although the current implementation adopted Revit and 3ds Max for carrying out the integration process, in order to realize the fully automated collection of needed coordinates and further enhance the interoperability and feasibility, we want to explore the use of IFC, an open standard that supports various tools for sharing, analyzing and editing BIM. Besides, in order to improve the reusability and accessibility, another further work is to expend this work to remote simulation with web-services.

REFERENCES

Ahmed, A., G. Wainer, and S. Mahmoud. 2010. "Integrating Building Information Modeling & Cell-DEVS Simulation." In *Proceedings of SimAUD 2010*. Orlando, FL.

AutoDesk. 2012. "Autodesk 3ds Max." Accessed June 15. http://usa.autodesk.com/3ds-max/.

AutoDesk. 2012. "Autodesk Maya." Accessed June 15. http://usa.autodesk.com/maya/.

AutoDesk. 2012. "Autodesk Revit Architecture." Accessed June 15. http://usa.autodesk.com/revit-architecture/.

Bentley Systems. 2012. "Bentley Architecture." Accessed June 15. http://www.bentley.com/en-US/Products/Bentley+Architecture/.

Boukerche, A., M. Zhang, and R. Pazzi. 2009. "An Adaptive Virtual Simulation and Real Time Emergency Response System." In *Intl. Conf. on Virtual Environments, Human-Computer Interfaces and Measurement Systems*. Hong Kong, China.

BuildingSMART. 2012. "BuildingSMART Data Model Standards, Industrial Foundation Classes (IFC)." Accessed April 15. http://buildingsmart-tech.org/.

Eastman, C., J-M. Lee, Y-S. Jeong, and J-K. Lee. 2009. "Automatic Rule-based Checking of Building Designs." *Journal of Automation in Construction*. Elsevier. 18(8):1011-1033.

Elbeltagi, E., T. Hegazy, and A. Eldosouky. 2004. "Dynamic Layout of Construction Temporary Facilities Considering Safety." *Journal of Construction Engineering and Management*. 130(4):534-541.

Graphisoft. 2012. "Graphisoft ArchiCAD." Accessed April 15. http://www.graphisoft.com/products/archicad/.

Hajjar, D., and S.M. AbouRizk. 2002. "Unified Modeling Methodology for Construction Simulation." *Journal of Construction Engineering and Management- ASCE*. 128(2):174-185.

Halpin, D.W. 1977. "CYCLONE: Method for Modeling of Job Site Processes." *Journal of the Construction Division- ASCE*. 103(3):489-499.

Ham, N.H., K.M. Min, J.H. Kim, Y.S. Lee, and J.J. Kim. 2008. "A Study on Application of BIM to Pre-design in Construction Project." In *3rd International Conference on Convergence and Hybrid Information Technology*. Busan, Korea.

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 WSC*, edited by S. Jain, R.R. Creasey, J. Himmelspach, K.P. White, and M. Fu. Phoenix, AZ. 3533-3544. New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Hardin, B. 2009. *BIM and Construction Management: Proven Tools, Methods, and Workflows*. Wiley.

Kincaid, J.P., J. Donovan, and B. Pettitt. 2003. "Simulation Techniques for Training Emergency Response." *International Journal of Emergency Management*. 1(3):238-246.

Lee, G., R. Sacks, and C. Eastman. 2006. "Specifying Parametric Building Object Behavior (BOB) for a Building Information Modeling System." *Automation in Construction*. 15(6):758-776.

Martinez, J.C. 2001. "EZStrobe - General-Purpose Simulation System Based on Activity Cycle Diagrams." In *Proceedings of the 2001 Winter Simulation Conference*. edited by B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer. Piscataway, Washington, DC. 1556-1564. New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Mihindu, S. and Y. Arayici. 2008. "Digital Construction through BIM Systems will Drive the Re-engineering of Construction Business Practices." In *International Conference on Visualization*, London, UK.

Pelechano, N., and A. Malkawi. 2008. "Evacuation Simulation Models: Challenges in Modeling High Rise Building Evacuation with Cellular Automata Approaches." *Journal of Automation in Construction.* Elsevier. 17(4):377-385

Ruppel, U., and P. Abolghasemzadeh. 2009. "BIM-based Immersive Evacuation Simulations." In *18th International Conference on the Application of Computer Science and Mathematics in Architecture and Civil Engineering*. Weimar, Germany.

VectorWorks. 2012. "Vectorworks Architect." Accessed June 15. http://www.nemetschek.net/architect/.

Wainer, G. 2009. *Discrete-event modeling and simulation: a practitioner's approach*. CRC/Taylor & Francis.

Wainer, G., E. Poliakov, J. Hayes, and M. Jemtrud. 2007. "A Busy Day at the SAT Building." In *Proceedings of AIS, Artificial Intelligence, Simulation and Planning*. Buenos Aires, Argentina.

Yang, L.Z., D.L. Zhao, J. Li, and T.Fang. 2005. "Simulation of Kin Behavior in Building Occupant evacuation based on Cellular Automaton." *Journal of Building and Environment.* Elsevier. 40(3):411-415.

Zeigler, B.P., H. Praehofer, and T.G. Kim. 2000. *Theory of Modeling and Simulation*. Academic Press.

Zhang, C., A. Hammad, T.M. Zayed, and G. Wainer. 2007. "Cell-based Representation and Analysis of Spatial Resources in Construction Simulation." *Journal of Automation in Construction.* Elsevier. 16(4):436-448.

AUTHOR BIOGRAPHIES

SIXUAN WANG is a Ph.D. student in Electrical and Computer Engineering with the Department of Systems and Computer Engineering at Carleton University. He received his B.Eng. and M.Eng. from Harbin Institute of Technology, China and a M.Sc. from University of Bordeaux 1, France. His research interests are discrete-event systems, distributed/web-based simulation, model-driven engineering and interoperability. His email is swang@sce.carleton.ca.

MICHAEL VAN SCHYNDEL is a Masters student in Biomedical Engineering with the Department of Systems and Computer Engineering, Carleton University. He received his B. Eng. in Biomedical and Electrical Engineering in 2011 from Carleton University. His research focuses on using discrete-event systems to model biological systems. His email is mvschynd@connect.carleton.ca.

GABRIEL A. WAINER (SMSCS, SMIEEE) is a Full Professor at the Department of Systems and Computer Engineering at Carleton University. He is the head of the Advanced Real-Time Simulation lab, located at Carleton University's Centre for advanced Simulation and Visualization (V-Sim). He has been the recipient of various awards, including the IBM Eclipse Innovation Award, SCS Leadership Award, and various Best Paper awards. He has been awarded the First Bernard P. Zeigler DEVS Modeling and Simulation Award, and the SCS Outstanding Professional Award (2011). Further information can be found at http://www.sce.carleton.ca/faculty/wainer.

VINU SUBASHINI RAJUS is a Ph.D. student in School of Interactive Arts and Technology, Simon Fraser University. She holds a M.Sc. in Digital Architecture from Anna University, India and a B. Arch. from Madras University. She is a part of the CAD research group at SIAT. Her research interest includes computational geometry, parametric modeling and sustainability. Her email is vrajus@sfu.ca.

ROBERT WOODBURY holds a B.Arch. from Carleton University (Lieutenant Governor's Silver Medal), M.Sc. and Ph.D. from Carnegie Mellon University. He was a faculty member in Architecture and the Engineering Design Research Center at CMU from 1982-1993, at Adelaide University (Australia) from 1993-2001, and is now at Simon Fraser University. From 2005-2008 he was Scientific Director of the Canadian Design Research Network and is currently the Director Art and Design Practice of Graphics, Animation and New Media Canada. His main interest is augmenting professional work by supporting state-space search in problem spaces. His email is robw@sfu.ca.